

## FATS/FATAR USERS MANUAL VERSION 4.8

FATS/FATAR is a package of magnetic tape utilities which includes:

- FATS (Fast Analysis of Tape Surfaces) provides certification and verification of tape volumes, erasing tapes and labeling tapes. FATS can be licensed separately.
- FATAR (Fast Analysis of Tape and Recovery) analyzes the contents of tape volumes and optionally copies the data to new tape volumes, including the ability to recover from errors on the input tapes, modify data during the copy, and rename and recatalog the output tape datasets. FATAR can be used to copy data from older media drives to newer media, such as IBM 3590 Magstar and StorageTek 9840.
- FATSCOPY uses the copying facilities of FATAR to automate stacking of datasets on tape to improve utilization of high-capacity tape volumes and simplify the migration to new media types. FATSCOPY is an optional component.

FATS, FATAR and FATSCOPY are proprietary products of:

INNOVATION DATA PROCESSING, INC.  
Innovation Plaza  
275 Paterson Avenue  
Little Falls, NJ 07424  
Tel: (973) 890-7300  
Fax: (973) 890-7147

and are available for license exclusively from INNOVATION DATA PROCESSING, INC.

FATS, FATAR and FATSCOPY are trademarks of INNOVATION DATA PROCESSING and are registered with the US Patent Office.

### EUROPEAN OFFICES:

#### France

Innovation Data Processing S.A.R.L.  
191 Avenue Aristide Briand  
94230 CACHAN  
Tel: 01-49-69-94-02 Fax: 01-49-69-90-98  
E-mail: frsupport@fdrinnovation.com  
frsales@fdrinnovation.com

#### Netherlands

Innovation Data Processing  
Brouwerstraat 8  
1315 BP Almere  
Tel: 036-534 1660 Fax: 036-533 7308  
E-mail: nlsupport@fdrinnovation.com  
nlsales@fdrinnovation.com

#### Germany

Innovation Data Processing  
International Ltd.  
Orleansstrasse 4a  
D-81669 München  
Tel: 089-489 0210 Fax: 089-489 1355  
E-mail: desupport@fdrinnovation.com  
desales@fdrinnovation.com

#### United Kingdom

Innovation Data Processing Ltd.  
Clarendon House  
125 Shenley Road  
Borehamwood, Herts, WD6 1AG  
Tel: 0208-905 1266 Fax: 0208-905 1428  
E-mail: uksupport@fdrinnovation.com  
uksales@fdrinnovation.com

(c) Copyright 1974, 1985, 1989, 1990, 1991, 1992, 1997, 1998, 1999, 2000, 2001, 2003  
(c) INNOVATION DATA PROCESSING, INC.

**Reproduction of this User Manual is prohibited except for licensed users for their internal use.**

A copy of this manual is provided on CD-ROM in IBM Bookmanager format for online use, and in Adobe PDF format for local printing.

### USA CONTACTS

Voice: 973-890-7300 Fax: 973-890-7147  
Home page: [www.innovationdp.fdr.com](http://www.innovationdp.fdr.com)

E-mail: Sales: [sales@fdrinnovation.com](mailto:sales@fdrinnovation.com)  
Technical Support: [support@fdrinnovation.com](mailto:support@fdrinnovation.com)

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## TABLE OF CONTENTS

Page No.

<b>1.0</b>	<b>INTRODUCTION</b>	
1.1	SUMMARY OF MODIFICATIONS .....	1.1
<b>10.0</b>	<b>FATS FUNCTIONAL DESCRIPTION</b>	
10.0	FATS FUNCTIONAL DESCRIPTION .....	10-1
<b>11.0</b>	<b>FATS TECHNICAL SUMMARY</b>	
11.0	FATS TECHNICAL SUMMARY .....	11-1
11.1	GENERAL .....	11-1
11.2	TAPE CERTIFICATION .....	11-2
11.3	TAPE VERIFICATION .....	11-5
11.4	TAPE LABELING .....	11-7
11.5	ERASING TAPES .....	11-8
11.6	TAPE MANAGEMENT SYSTEMS .....	11-9
11.7	REPORT FORMATS .....	11-10
11.8	SECURITY .....	11-12
<b>12.0</b>	<b>FATS GUIDLINES</b>	
12.0	FATS GUIDELINES .....	12-1
12.1	CAUSES OF TAPE ERRORS .....	12-2
12.2	OPERATING SYSTEM VS. FATS ERRORS .....	12-3
12.3	VARIATIONS IN FATS RESULTS .....	12-4
12.4	ANALYSIS OF CERTIFICATION RESULTS .....	12-5
12.5	ANALYSIS OF VERIFICATION RESULTS .....	12-6
12.6	RECOMMENDATIONS FOR FATS USAGE .....	12-7
12.7	QUESTIONS AND ANSWERS .....	12-8
<b>13.0</b>	<b>FATS EXECUTION JCL</b>	
13.0	FATS EXECUTION JCL .....	13-1
13.1	OPERATOR CONTROL .....	13-3
<b>14.0</b>	<b>FATS CONTROL STATEMENTS</b>	
14.0	FATS CONTROL STATEMENTS .....	14-1
14.1	GENERAL .....	14-1
14.2	OPERATION KEYWORDS .....	14-2
14.3	OPERANDS .....	14-3
14.3.1	RETRY/ERROR OPERANDS .....	14-3
14.3.2	VOLUME MOUNTING OPERANDS .....	14-4
14.3.3	LABELING OPERANDS .....	14-6
14.3.4	PRINT OPERANDS .....	14-7
14.3.5	CONTROL OPERANDS .....	14-8
14.3.6	MISCELLANEOUS OPERANDS .....	14-8
14.3.7	ANALYZE OPERANDS .....	14-9
<b>16.0</b>	<b>FATS EXAMPLES OF USAGE</b>	
16.0	FATS EXAMPLES OF USAGE .....	16-1
16.1	EXAMPLES .....	16-2

## TABLE OF CONTENTS

Page No.

<b>20.0</b>	<b>FATAR FUNCTIONAL DESCRIPTION</b>	
20.0	FATAR FUNCTIONAL DESCRIPTION .....	20-1
<b>21.0</b>	<b>FATAR TECHNICAL SUMMARY</b>	
21.0	FATAR TECHNICAL SUMMARY .....	21-1
21.1	GENERAL .....	21-1
21.2	INPUT TAPE PROCESSING .....	21-2
21.3	MODIFICATION AND SCANNING .....	21-7
21.4	OUTPUT TAPE PROCESSING .....	21-8
21.5	TAPE MANAGEMENT SYSTEM SUPPORT .....	21-12
21.6	SECURITY .....	21-13
<b>22.0</b>	<b>FATAR EXECUTION JCL</b>	
22.0	FATAR EXECUTION JCL .....	22-1
<b>23.0</b>	<b>FATAR CONTROL STATEMENTS</b>	
23.0	FATAR CONTROL STATEMENTS .....	23-1
23.1	GENERAL .....	23-1
23.2	ANALYZE STATEMENT .....	23-2
23.2.1	INPUT TAPE OPERANDS .....	23-3
23.2.2	ERROR RECOVERY OPERANDS .....	23-6
23.2.3	OUTPUT TAPE OPERANDS .....	23-7
23.2.4	PRINT CONTROL OPERANDS .....	23-10
23.2.5	MISCELLANEOUS OPERANDS .....	23-12
23.3	MODIFICATION/SCAN STATEMENTS .....	23-13
23.3.1	PRINT STATEMENT .....	23-13
23.3.2	DROP STATEMENT .....	23-14
23.3.3	KEEP STATEMENT .....	23-14
23.3.4	REPLACE STATEMENT .....	23-15
23.3.5	SCAN STATEMENT .....	23-16
23.3.6	TABLE STATEMENT .....	23-16
23.3.7	RENAME STATEMENT .....	23-17
23.4	MODIFICATION/SCAN OPERANDS .....	23-19
<b>24.0</b>	<b>FATAR SPECIAL CONSIDERATIONS</b>	
24.0	FATAR SPECIAL CONSIDERATIONS .....	24-1
24.1	VARIABLE SPANNED RECORDS .....	24-1
24.2	RECOVERING DATA FROM OVERWRITTEN TAPES .....	24-3
24.8	FATAR REPORT EXAMPLES .....	24-6
<b>26.0</b>	<b>FATAR EXAMPLES OF USAGE</b>	
26.0	FATAR EXAMPLES OF USAGE .....	26-1
26.1	INDEX TO EXAMPLES .....	26-2
26.2	EXAMPLES .....	26-3

# TABLE OF CONTENTS

Page No.

<b>30.0</b>	<b>FATSCOPY FUNCTIONAL DESCRIPTION</b>	
30.0	FATSCOPY FUNCTIONAL DESCRIPTION .....	30-1
<b>31.0</b>	<b>FATSCOPY EXECUTION JCL</b>	
31.0	FATSCOPY EXECUTION JCL .....	31-1
<b>32.0</b>	<b>FATSCOPY CONTROL STATEMENTS</b>	
32.0	FATSCOPY CONTROL STATEMENTS .....	32-1
32.1	GENERAL .....	32-1
32.2	COPY/SIM STATEMENT .....	32-2
32.3	SELECT/EXCLUDE STATEMENT .....	32-7
32.4	DATASET NAME FILTERS .....	32-11
32.5	FATSCOPY SPECIAL CONSIDERATIONS .....	32-14
<b>33.0</b>	<b>FATSCOPY EXAMPLES</b>	
33.0	FATSCOPY EXAMPLES .....	33-1
<b>40.0</b>	<b>FATS AND FATAR MESSAGES AND CODES</b>	
40.0	FATS AND FATAR MESSAGES AND CODES .....	40-1
40.1	RETURN CODES .....	40-1
40.2	ABEND CODES .....	40-1
40.3	CONSOLE MESSAGES .....	40-2
40.4	PRINT MESSAGES .....	40-5
40.5	FATS ACTION MESSAGES .....	40-31
<b>90.0</b>	<b>FATS AND FATAR INSTALLATION</b>	
90.0	FATS AND FATAR INSTALLATION .....	90-1
90.1	LOADING THE FATS/FATAR LIBRARIES FROM THE DISTRIBUTION TAPE ..	90-2
90.2	ALTERNATE MANUAL INSTALLATION .....	90-7
90.3	IMPLEMENTING THE TAPE MANAGEMENT INTERFACE .....	90-8
90.4	CUSTOMIZING FATS AND FATAR .....	90-9
90.5	INSTALLING THE ISPF DIALOG .....	90-10
90.6	VIEWING/PRINTING THE SOFTCOPY DOCUMENTATION .....	90-11
<b>93.0</b>	<b>FATS/FATAR ISPF INTERFACE</b>	
93.0	ISPF INTERFACE INTRODUCTION .....	93-1
93.1	FATS ISPF INTERFACE .....	93-5
93.2	FATAR ISPF INTERFACE .....	93-9
93.3	FATAR EXTENDED FUNCTIONS ISPF INTERFACE .....	93-12

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 1.1 SUMMARY OF MODIFICATIONS

**SUMMARY OF MODIFICATIONS FOR V4.8.30****FATSCOPY  
SELECT BY  
VOLSER**

FATSCOPY can now copy specified tape volume serials. If SELECT ALLDSN,VOL=vvvvvvv is specified, the tape will be selected using tape management information, not the catalog. All datasets on the input tape will be copied. If the tape volser is part of a multi-volume tape set, all tapes in the set (and all datasets on those tapes) will be copied.

By default, FATSCOPY will use one output tape (or output tape set if it is too big for a single tape) for each input tape or tape set. However, you can request that the input tape files will be stacked on a single output tape or tape set.

**CAT=ONLY**

The new CAT=ONLY operand, accepted by FATAR and FATSCOPY, will catalog output tape files only if the input dataset was cataloged. The existing catalog entry must point to the input tape volser with the proper file number. If the catalog entry does not match, or the input dataset is not cataloged, the output dataset will not be cataloged.

CAT=ONLY is now the default for FATSCOPY. However, when you are selecting input datasets from the catalog, they will naturally be cataloged, so this will work the same as the previous default of CAT=RECAT. Only when selected tapes by volser will CAT=ONLY be effective.

The default in FATAR remains CAT=NO

**SUMMARY OF MODIFICATIONS FOR V4.8.20****3590-H  
SUPPORT**

3590-H (triple density) tape drives are now supported.

**VTS AND  
COPYCROSS  
SUPPORT**

Tape volumes in an IBM VTS (virtual tape system) will now be identified as device type VIRTUAL. The capacity and used percentage for a VTS virtual volume will be reported.

CopyCross is a virtual tape software product from EMC which emulates tape devices and stores the virtual tapes on EMC Symmetrix disks. Tape volumes in a CopyCross system will now be identified as device type VIRTUAL.

**FATSCOPY  
ENHANCE-  
MENTS**

- File statistics (bytes, blocks, etc.) will be printed for each file copied.
- FDRABR backup tape files and DFSMSHsm backup and migration tape files will be automatically excluded by matching the tape dataset name against various masks. The default masks will exclude the default names used by each product. FDRABR customers who have change the ABRINDEX from FDRABR to another value can override this index. DFSMSHsm customers who have overridden the default names can specify alternate name masks.
- Datasets can be selected based on their size in mega-bytes (MB), if that information is available from the tape management system.
- Copy the "volume assigned" date and time when DFSMSrmm is the tape management system.
- Copy the "creating program name" when CA-1 is the tape management system.
- Selection can be limited to tape datasets cataloged to EMC CopyCross virtual volumes.
- In a simulation run, the capacity of the output tapes that will be used can be specified. The report will display the estimated number of volumes of that capacity required to hold the data selected by the simulation.
- A DCB ABEND exit was added to allow FATSCOPY to recover from various OPEN ABENDs on selected input datasets and continue copying other datasets.

## 1.1 CONTINUED . . .

**SUMMARY OF MODIFICATIONS FOR V4.8.00**

**FATSCOPY** FATSCOPY is a new, optional component of the FATS/FATAR family. It uses the copying facilities of FATAR to automate copying and stacking files on tape. Files to be copied can be selected from your system catalogs by dataset name mask, volume serial range or prefix, and/or tape device type. The selected files will be dynamically allocated and copied to consecutive files on the output tapes.

FATSCOPY can be used to automate the conversion from one tape device type to another, such as conversion from lower-capacity cartridges (such as 3480 and 3490E) to high-capacity cartridges (such as IBM 3590 Magstar and StorageTek 9840). It will automatically stack the copied files on the output tapes, using fewer cartridges to hold the copied data.

FATSCOPY can also be used to periodically collect tape files on tapes by themselves and stack them on output tapes, using the tapes more efficiently.

In this release, FATSCOPY supports the CA-1 and DFSMSrmm tape management systems. It will extract information for copied tape files from those systems, and will optionally copy that information to the associated output tape files. It will also optionally scratch copied input tapes under those tape management systems.

**FATAR COPY STATEMENT** FATAR now supports a main statement of COPY in addition to ANALYZE. COPY is the same as ANALYZE, except that it implies that the input tape is being copied, so a TAPEOUT DD statement is required. Either COPY or ANALYZE can be used when copying a tape with FATAR. All ANALYZE operands are supported on COPY.

This change was made simply to make it clearer when a FATAR jobstream was being used to analyze an input tape or to copy a tape.

**256K BLOCK SUPPORT** FATAR can now read and copy tapes with blocks up to 256K in length. Blocksizes up to 256K are supported by OS/390 2.10 and z/OS.

**CERTIFICATION OF HIGH-CAPACITY TAPES** Although FATS supports certification (WRITE) of IBM 3590 Magstar and StorageTek 9840 and T9940 tape cartridges, the certification of an individual cartridge will take a long time (20-60 minutes, depending on tape model and capacity). With the increased reliability of high-capacity drives and cartridges compared to earlier technology, it is questionable whether it is appropriate to routinely certify all new Magstar cartridges. Innovation suggests that you may want to certify high-capacity cartridges only when problems have been identified on a cartridge.



**10.0 FATS FUNCTIONAL DESCRIPTION**

<b>WHAT IS FATS?</b>	FATS (Fast Analysis of Tape Surfaces) is a special-purpose magnetic tape utility, which can be executed under MVS/ESA and OS/390. FATS is used to perform many of the functions for which a tape librarian is responsible in most shops, such as certifying usability of new or scratch tapes, verifying readability of tapes, erasing data from tape, and labeling tapes. However, these operations are done on the actual tape drives where the tapes will be used rather than on expensive stand-alone hardware devices. This results in a savings of money and of time, since FATS can perform as many concurrent operations as there are tape drives available.
<b>WHAT IS FATS USED FOR?</b>	<p>FATS has several specific applications, including:</p> <ul style="list-style-type: none"> <li>– Certifying the usability of tapes, both new and old, by writing a pattern on the full length of the tape (this is equivalent to the function of hardware tape certifiers). This helps to avoid write data errors at application run time by identifying potentially defective tapes.</li> <li>– Verifying the readability of critical or archival data tapes, to avoid read data errors at application run time. Tapes with read errors may be recovered using FATS' companion product FATAR.</li> <li>– Labeling tapes with specified volume serials. This can be done by itself, or in combination with a certification process.</li> <li>– Erasing data from a tape, to insure that confidential information is not left on a tape (as when a tape volume is being sold or shipped to another location). This is similar to stand-alone tape degaussers, but is superior since tape serials can be verified and preserved.</li> <li>– Interfacing with its companion product FATAR to perform detailed data analysis and copying of tapes.</li> </ul> <p>Regular use of the certification and verification functions of FATS can significantly reduce the number of application abends due to data checks. In addition, many tape data checks are recoverable by operating system error recovery, but cause significant run time degradation while this recovery is taking place; FATS can reduce this degradation.</p>
<b>WHO USES FATS?</b>	FATS is normally used by tape librarians and operations personnel but may be used by systems and applications programmers.
<b>TAPE OPERATIONS</b>	Depending on the command statements provided, FATS will perform any of its supported functions on from 1 to 9 tape drives simultaneously. The same operation could be performed on all of the allocated tape drives, or, if desired, different functions could be done on different drives. FATS uses an internal task control so that all of these operations can proceed independently of one another. By the use of FATS parameters, FATS can be instructed to request multiple tapes on a drive, calling either for scratch tapes or specific volume serials, so that the same operation can be performed on many tapes using one tape drive. FATS will terminate when instructed by the operator or when all requested functions are completed.

**10.0 CONTINUED . . .****SUPPORTED  
DEVICES**

FATS will function on any IBM or IBM-compatible tape drive supported by the host MVS system including:

- IBM 3420 and 3422 9-track round (reel-to-reel) drives up to 6250 BPI and equivalent drives from other vendors. By default, multi-density drives certify tapes at the highest density they are capable of, although this can be overridden if required.
- IBM 3480 and 3490 tape cartridge systems (18 track) and equivalent drives from other vendors such as StorageTek
- IBM 3490E tape cartridge systems (36 track) and equivalent drives from other vendors such as StorageTek.
- IBM 3590 (Magstar) tape cartridge systems (128 track) in either native or 3490E emulation mode.
- IBM 3590E (Magstar) tape cartridge systems (256 track) including the extended length cartridge.
- IBM 3590H (Magstar) tape cartridge systems (384 track) including the extended length cartridge.
- StorageTek Timberline 9490 (3490E-compatible) cartridge systems, including those supporting quadruple-length EETape cartridges.
- StorageTek 9840 and T9940 tape cartridge systems (emulates 3490E or 3590).
- StorageTek Redwood SD-3 tape cartridge systems (may emulate either 3490E or 3590)
- FATS may be used to label or read tapes contained in a Virtual Tape System (such as the IBM 3494 VTS). However, there is no point in certifying or erasing VTS tapes since the tape data is buffered on an internal disk and no real tape surface is involved.

**PROCESSING  
SPEED**

The speed at which FATS can process a tape is dependent on three variables: the speed of the tape drive, the contention for the tape channel and control unit, and the quality of the tapes being tested. When testing tapes suspected of being exceptionally poor, a low permanent error retry level can be specified to accelerate processing. FATS may also contend with itself if multiple tapes being tested concurrently are on the same tape control unit or channel; when processing more than three tapes in one FATS job (except for ERASE functions), it is wise to process multiple tapes using more than one channel and control unit, if available.

When certifying or erasing a single tape, you can expect elapsed times similar to:

Certify a 3480 cartridge	2.5 minutes
Erase a 3480 cartridge	1.8 minutes
Certify a 3490E standard (500 ft) cartridge	4.8 minutes
Erase a 3490E standard cartridge	3.6 minutes
Certify a 3490E Enhanced (1100 ft) cartridge	10.0 minutes
Erase a 3490E Enhanced cartridge	7.2 minutes
Certify a StorageTek Timberline standard cartridge	1.9 minutes
Certify a StorageTek Timberline Enhanced cartridge	3.2 minutes
Certify a 3590 (Magstar) 10GB cartridge	26.1 minutes
Erase a 3590 (Magstar) cartridge	19.6 minutes

Erase times will be independent of other tape activity, but certify times will increase if there is other tape activity (including other FATS or FATAR activities) on the same tape channel or control unit. Read times will vary depending on the amount of data and blocksize on the tape and cannot be estimated.

**11.0 FATS TECHNICAL SUMMARY****11.1 GENERAL****THE FATS  
PROGRAM**

FATS (Fast Analysis of Tape Surfaces) is a utility program for certifying, verifying, labeling, and erasing magnetic tapes.

FATS requires 256K of memory. If FATS' companion product FATAR is invoked as a sub-task (via the ANALYZE(n) operation), then the memory requirements of FATAR (detailed elsewhere in this manual) must be added to that of FATS for each FATAR sub-task.

FATS must be linkedited and executed as an authorized program.

**CONTROL AND  
REPORT I/O**

FATS accepts its control statement input from DD statement SYSIN. The control statements are 80-byte records, of which only columns 1 to 71 may contain information.

Report output is directed to several print datasets. DD statement SYSPRINT is required, and will contain general messages and control statement listings. Up to 9 optional SYSPRINx DD statements may be included (where "x" is a digit from 1 to 9 and corresponds to the 9 tape drives used by FATS); if present, detail reports on the results from the associated tape drive will be printed on SYSPRINx, but if omitted, the detail reports will go to SYSPRINT. DD statement SUMMPRT is also optional; if present, it will receive a summary report on each tape volume processed by FATS; if absent, the summary report will go to the associated SYSPRINx or to SYSPRINT.

**TAPE OPEN  
PROCESSING**

FATS always opens every tape with a standard data management OPEN, so that label processing will be performed (unless bypassed by JCL or control statement options), and tape management and security systems, if present, will be invoked. So, for all FATS operations, you have assurance that the tapes mounted are the proper tapes and that they may be written on or read from as long as you do not bypass these checks yourself.

If label processing is bypassed (LABEL=(,BLP) in JCL or the BLP control statement option), the FATS option VALIDATE= may be used to check the volume serial and expiration date on tapes used by FATS.

**FATS  
CONTROLS**

The system operator or tape librarian can exercise control over FATS by optionally replying to an outstanding WTOR (write-to-operator-with-reply) or by issuing a console STOP or MODIFY command. Use of WTOR=YES operand (the default) will cause a WTOR to be issued to the tape pool console (route code 3) when FATS is executing. The operator or librarian can terminate FATS at any time by replying to this message; FATS can also be instructed to abandon processing of a particular tape (if, for example, the tape is too damaged to mount). Alternatively, by using the MODIFY=YES parameter (default is MODIFY=NO), the operator or librarian may use console STOP and MODIFY commands to gain the same control over FATS.

**ISPF PANEL  
SUPPORT**

Extensive ISPF Panels are available to perform many of the FATS/FATAR functions like labeling, certifying and copying tapes. See Section 93 on "How to Use the ISPF Panels."

## 11.2 TAPE CERTIFICATION

**OPERATING  
SYSTEM  
RECOVERY**

When a WRITE command is issued to a tape drive, the tape hardware reads back the data just written to verify it and indicates the success or failure of the WRITE. If the readback fails a WRITE DATA CHECK is indicated and a recovery action is initiated which generally involves:

- Backspace over the bad record.
- Erase a section of tape
- Retry the failing WRITE command.

This will normally be done until the write completes normally or a retry limit is reached. For round tapes, this recovery is done by operating system software, but in tape cartridge systems, the recovery operations are done by the tape control unit without assistance from the operating system. The effect is the same in either case.

If all retries fail, a permanent WRITE DATA CHECK is indicated, and the operating system will write an I/O error message on the console and report the error back to the requesting application program, which usually results in abnormal termination of that job.

Even if the recovery is eventually successful, all of the repositioning and retrying can impact the performance of tape channels and control units. In the worst case, many inches of tape may be erased to recover from a very long error; a large number of these errors can significantly reduce the usable length of a tape and elongate the run time of the job.

Permanent (unrecoverable) and temporary (eventually successful) data check counts are accumulated by the operating system and tape hardware and are recorded both on "SYS1.LOGREC" (EREP) and on SMF (if you are keeping SMF type 21 records which are "ESV" – Error Statistics by Volume). However, all of the above RETRY operations are recorded as only one permanent or temporary error even though the WRITE may have failed at up to 16 different locations on the tape. No indication is given of the length of an error.

The purpose of operating system error recovery is to attempt to let an application job complete successfully despite WRITE errors on output tapes, regardless of the cost. This is an excellent goal, but greater reliability and performance could be achieved if the errors could be avoided in the first place.

**TAPE  
CERTIFICATION**

The FATS certification function identifies error locations on a tape before it is used by an application program.

The technique used is simple. FATS writes large blocks of patterned data. The hardware readback function of the tape drive is used to detect blocks which were not written successfully (data checks). FATS will inhibit the operating system and tape hardware from automatically attempting to recover from such data checks, since we want to identify and report on them.

When data checks occur, FATS will backspace over the bad block and retry the WRITE in the **same location**. If a user-specified RETRY limit is exceeded for errors at one location (default is 10), the error is considered permanent and is probably due to some tape media defect. If the WRITE is completed successfully before the RETRY limit is reached, it is considered a temporary data check (this may be due to debris on the tape which was dislodged during the retry process).

FATS will report, in the associated detail report for the tape, the location (in feet from the beginning of the tape) of any temporary or permanent error. The length of permanent errors in inches will also be reported. If permanent errors occur at consecutive locations, FATS will report the cumulative length of the error. The summary report for the tape will report the total number of permanent errors, and temporary error totals by number of RETRIES. Lengths and locations can optionally be reported in metric units (meters and centimeters).

By reporting the number, length, and location of all errors, FATS provides sufficient information to make an informed decision on whether a given tape should be put into use, discarded, or repaired (more detail on this is given in the section "FATS GUIDELINES").

## 11.2 CONTINUED . . .

**TAPE  
CERTIFICATION  
(continued)**

The location of errors is calculated by FATS in feet or meters since the beginning of the tape is a function of the number and length of FATS records written, and the number of inter-record gaps. FATS assumes the nominal inter-record gap size for each tape drive and density as documented in IBM tape drive hardware manuals. However, the actual length of this gap can vary from drive to drive, so a given tape certified on two different drives may report errors at slightly different locations. FATS will also report the total length of the tape, but this may also vary due to gap variations.

Tape data checks may be caused by tape drive equipment checks, by dirty tape read/write heads, by tape surface damage (or improper manufacture), or by dirt or other contaminants on the surface. Equipment failure is the least likely and will usually show up as errors across the entire length of a number of tapes. Errors caused by dirt are often transitory; they may move or disappear if the tape drive picks up the dirt and re-deposits it in a different location.

When certifying new tapes, FATS can be instructed to mount a range of sequentially numbered tapes with simple control statements.

**IDRC ON  
CARTRIDGE  
DRIVES**

Although most cartridge drives (except some early 3480 drives) are capable of hardware compression of tape data using a feature known as IDRC (Improved Data Recording Capability), IDRC will always be suppressed during certification, even if it is called by the tape JCL (TRTCH=COMP) or by the system default. IDRC compression is not relevant to a certification process and will simply make the certification take longer.

**LABELING  
TAPES**

FATS can label new tapes or relabel existing tapes during the certification process. Labels can be IBM standard or ANSI V3 labels. FATS can also create unlabeled tapes. See Section 11.4 for more information.

**CERTIFICATION  
OF TAPE  
CARTRIDGES**

Tape cartridges have resulted in a significant reduction in the number of data checks compared to round tapes, yet data checks do occur, and when they do they tend to be more severe than those on round tapes. FATS is an excellent tool to certify new cartridges and insure that the existing cartridges in the scratch pool that have potential for errors are **identified** and **eliminated** from the tape library.

**If bad cartridges are introduced into your tape pool, significant cost and time is required to identify them and to copy the data to a new set of tapes.**

**CERTIFICATION  
OF LARGE  
CAPACITY  
TAPES**

Although FATS supports certification (WRITE) of IBM 3590 Magstar and StorageTek 9840 and T9940 tape cartridges, the certification of an individual cartridge will take a long time (20-60 minutes, depending on tape model and capacity). With the increased reliability of high-capacity drives and cartridges compared to earlier technology, it is questionable whether it is appropriate to routinely certify all new Magstar cartridges. Innovation suggests that you may want to certify high-capacity cartridges only when problems have been identified on a cartridge.

## 11.2 CONTINUED . . .

**ISPF PANELS  
FOR  
CERTIFICATION**

Below are examples of the FATS ISPF panels for certification of new and existing tapes. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

**FATS CERTIFY  
NEW TAPES  
(OPTION A)**

```

----- FATS CERTIFY NEW TAPES -----
COMMAND ==>

TAPE1 DD Dsname= BAB.FATSOUT.JOBD          <
        Unit  =( TAPE                      )
        Label =( ,BLP,EXPDT=98000          )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                       5:  -----  6:  -----  7:  -----  8:  -----
                                       9:  ----- 10:  ----- 11:  ----- 12:  -----
                                       13: ----- 14:  ----- 15:  ----- 16:  -----
                                       17: ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN  - Maximum number of volumes :  -- (yes/no)
NOLABEL  - Create no label (NL) tapes:  NO (yes/no)
ANSI     - Create ANSI labeled tapes :  NO (yes/no)

Display more FATS Operands?: NO (yes/no)

```

**FATS CERTIFY  
EXISTING  
SCRATCH  
TAPES  
(OPTION B)**

```

----- FATS CERTIFY EXISTING TAPES -----
COMMAND ==>

TAPE1 DD Dsname= BAB.FATSOUT.JOBD          <
        Unit  =( TAPE                      )
        Label =( ,SL                      )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                       5:  -----  6:  -----  7:  -----  8:  -----
                                       9:  ----- 10:  ----- 11:  ----- 12:  -----
                                       13: ----- 14:  ----- 15:  ----- 16:  -----
                                       17: ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN  - Maximum number of volumes :  -- (yes/no)
NOLABEL  - Create no label (NL) tapes:  NO (yes/no)
ANSI     - Create ANSI labeled tapes :  NO (yes/no)

Display more FATS Operands?: NO (yes/no)

```

## 11.3 TAPE VERIFICATION

**TAPE  
VERIFICATION**

When reading tape, the operating system or tape control unit will retry any read data check many times by backspacing and re-reading the block. Cleaning actions (attempts to dislodge any dirt by moving the tape rapidly back and forth) are done periodically. In the end, if the tape drive is unable to read the block successfully, the error is reported to the application program which usually abnormally terminates.

FATS read verification may be used on critical or archival tapes before they are required by an application job to verify their readability. FATS will read each block on the input file, and will attempt to re-read each data check block a user-specified number of times (10 retries by default) to see if the error will go away with repeated reads.

FATS will report in its detail report the location (in feet or meters), file number, and record number of any temporary or permanent read data check encountered. Much like the certification function, FATS will report the length of any permanent error, and the total length of any consecutive permanent errors. The summary report for the verification will show the number of permanent errors, and the number of temporary errors by the number of retries. Lengths and locations can optionally be reported in metric units (meters and centimeters).

FATS performs its verification by reading without actually transferring the data read to the CPU; this is sufficient to identify data checks. For a more detailed look at the data causing the errors, or to attempt to correct or bypass the errors, FATS' companion product FATAR may be used. FATAR can be executed directly, or as a sub-task of FATS. FATAR is more appropriate for verifying tapes where the number of data files on the tape is unknown.

**IDRC ON  
CARTRIDGE  
DRIVES**

Most cartridge drives (except some early 3480s) support hardware data compaction through a feature called IDRC (Improved Data Recording Capability). FATS verification will automatically support tapes created with IDRC; no special parameters or JCL is required. A message in the FATS detail report will indicate when an IDRC-compacted tape file is read (called 3480XF or 3490E format). For IDRC tapes, compaction information read from the tape will be used to calculate accurate locations of errors and files; the percentage of data compaction (original data size vs. compacted data size) is also reported.

**TAPE  
COMPATIBILITY**

In most cases, tapes written on one type of tape drive cannot be read on another, even when the tape types are similar and allow the tape to be mounted on the wrong type of drive. An attempt to read a tape on a drive which is not capable of reading its format will receive an error; in most cases it includes the text "NOT CAPABLE". Also, an attempt to read a 3480 cartridge written in IDRC format on a 3480 drive which does not have the IDRC feature gets a similar error.

Exceptions include:

- cartridges written on an IBM 3480/3490 drive (or compatibles) can be read on an IBM 3490E drive.
- cartridges written on an IBM 3590 drive can be read on an IBM 3590E drive
- many round (3420-type) tape drives are capable of reading tapes written in several densities.

Note that tapes created on an IBM Magstar (3590) tape drive in 3490E emulation mode may be read or written only on a Magstar, and may **not** be read or written on a standard 3490E drive. All Magstar tapes are in the same format which can be handled only by a Magstar drive; Magstar cartridges may not even be mounted on other cartridges drives, even though the cartridges are similar in size and appearance.

Likewise, tapes created on a StorageTek Redwood SD-3 or StorageTek 9840 cannot be read or written on any other type of drive, even though they emulate 3490E or 3590. StorageTek Timberline 9490 drives are fully compatible with IBM 3490E drives (except for StorageTek quadruple-length EETape cartridges).

## 11.3 CONTINUED . . .

**ISPF PANEL FOR  
VERIFICATION**

Below is an example of the FATS ISPF panel for verification of active tapes. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

**FATS VERIFY  
(READ)  
OF EXISTING  
DATA  
ON TAPE  
(OPTION F)**

```

----- FATS VERIFY (READ) TAPES -----
COMMAND ==>
TAPE1 DD Dsname=  FATS                               <
        Unit  =( 3480 )
        Label =( ,BLP,EXPDT=980000 )
VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                       5:  -----  6:  -----  7:  -----  8:  -----
                                       9:  ----- 10:  ----- 11:  ----- 12:  -----
                                       13: ----- 14:  ----- 15:  ----- 16:  -----
                                       17: ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :
FILES    - Number of Physical files to be read: 3

```



## 11.4 TAPE LABELING

FATS labels a tape by writing an IBM standard (SL) or ANSI V3 (AL) label set.

FATS can write labels during certification or erase of a tape, and can also label tapes as a separate function (at considerable savings in elapsed time). Because FATS can request mounting of a large number of volumes, it is ideal for initializing a sequence of new tapes, or re-labeling old tapes with new serials.

The serial number can be specified in FATS control statements, in JCL or via the operator's console. However, for ease in certifying tapes which are already labeled, the FATS default option SAVLAB will read the existing volume serial from the tape being certified and re-label the tape with it, thus "saving" the volume serial.

FATS can also create unlabeled tapes, writing only a tape mark at the beginning of the tape.

When labeling new tapes, FATS can be instructed to mount a range of sequentially numbered tapes with simple control statements.

**FATS LABEL SET**

When labeling tapes, either directly via the LABEL function, or as an option of the WRITE or ERASE functions, FATS will write the following records:

VOL1 label – with the specified volume serial, and optional OWNERID
HDR1 label *
HDR2 label – showing RECFM=U and BLKSIZE=32760
Tape Mark
Tape Mark – empty data file
EOF1 label *
EOF2 label – showing RECFM=U and BLKSIZE=32760
Tape Mark
Tape Mark – end of tape

\* the HDR1 and EOF1 labels will contain the dataset name that appeared in the TAPEX DD statement, except that if LABEL=EXPDT=98000 also appeared on that DD statement, the dsname will consist of 17 "0"s (zero character) so that the tape will be immediately acceptable as a scratch tape for tape management systems. The creation and expiration dates are set to the current date.

The labels will be IBM standard, except that if the ANSI operand is specified, they are ANSI V3 standard labels.

**NOTE:** the labels written by FATAR differ from IEHINITT and some other tape labeling products, which only write a VOL1, HDR1, and one Tape Mark. Although this minimal label set is valid, it may cause some tape mapping software to attempt to read past the tape mark and get a read error.

**ISPF PANEL FOR LABELING**

Below is an example of the FATS ISPF panel for labeling of new tapes or relabeling old tapes with new serial numbers. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

**FATS LABELING OF NEW TAPES (OPTION C)**

```

----- FATS LABEL NEW TAPES -----
COMMAND ==>

TAPE1 DD Dsname= FATS                                <
        Unit  =( TA80                                )
        Label =( ,BLP,EXPDT=98000                      )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                       5:  -----  6:  -----  7:  -----  8:  -----
                                       9:  ----- 10:  ----- 11:  ----- 12:  -----
                                       13: ----- 14:  ----- 15:  ----- 16:  -----
                                       17: ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :  N0- (yes/no)
NOLABEL - Create no label (NL) tapes :  NO (yes/no)
ANSI     - Create ANSI labeled tapes :  NO (yes/no)

```

## 11.5 ERASING TAPES

**ERASING TAPES**

All tape drives support a "Data Security Erase" feature, which can be used to erase data on a tape at high-speed without tying up the tape control unit. In other words, this erase function can clear all data from a tape without impacting other jobs using tape drives on the same tape control unit. The erase takes place at maximum tape write speed (which varies by tape drive model).

"Data security Erase" is not supported by any standard IBM software. However, FATS fully supports the erase function. When invoked, FATS ERASE will write labels on the tape (if indicated), and then start the hardware erase function which will continue without intervention from the CPU or control unit until the end-of-tape is reached. For tape cartridge systems, erase involves writing a random pattern on the tape for complete security. For round tapes, the tape drive's erase head is used to erase the tape to "unrecorded" status.

By default, ERASE will preserve the existing volume serial on the tape, performing a FATS label function before starting the hardware erase (see Section 11.4). Optionally, you can relabel it with a new serial number, or create an unlabeled tape.

This ERASE function may be important to installations which are upgrading to new tape technology and are planning to sell or discard many of their old tapes. **ERASE may be used to insure that all company confidential data is gone from those tapes before they leave the site.**

Note that the ERASE function, because it is a single very long I/O operation, may receive IBM message

**IOS071I uuu DEVICE END MISSING**

This is normal and can be ignored. If these messages become annoying, you can eliminate them by increasing the MIH (Missing Interrupt Handler) timeout value for the tape drives (this may be modified in the IECIOSxx member of PARMLIB or by a SETIOS console command). Tape cartridge drives will display the message "ERASE" or "ERASING" while the ERASE operation is in progress.

**WARNING: This function will erase all data on a tape/cartridge with no possibility of recovery. Use caution when using ERASE.**

**ISPF PANEL FOR  
ERASE**

Below is an example of the FATS ISPF panel for erasing tapes. See Section 93 for more information on the use of ISPF with FATS. Option numbers shown are options on the FATS/FATAR main menu.

**FATS ERASE  
FUNCTION  
(OPTION E)**

```

----- FATS ERASE OLD DATA -----
COMMAND ==>

TAPE1 DD Dsname= FATS                               <
        Unit  =( TA80                               )
        Label =( ,SL                                )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                         5:  -----  6:  -----  7:  -----  8:  -----
                                         9:  ----- 10:  ----- 11:  ----- 12:  -----
                                        13:  ----- 14:  ----- 15:  ----- 16:  -----
                                        17:  ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :  YES (yes/no)
SAVLAB  - Preserve volume serial?    :  YES (yes/no)
NOLABEL - Create no label (NL) tapes: NO  (yes/no)
ANSI    - Create ANSI labeled tapes : NO  (yes/no)

```

**11.6 TAPE MANAGEMENT SYSTEMS****TAPE  
MANAGEMENT  
SYSTEMS**

Although FATS has no formal interface to any tape management system, it is designed to be compatible with them. FATS takes the following actions for the benefit of tape management systems:

- Tapes will always be opened with the label type indicated in the JCL (unless the FATS BLP operand is used). If tapes are opened as labeled, the tape management system knows the volume serial of the tape and can approve or disapprove its use. If they are opened as unlabeled, an operator response may be required to provide the volume serial for the tape management system.
- When tapes are opened for output (certification, labeling or erasure), the expiration date of the dataset will be set to the current date, so that the dataset will be immediately available as a scratch tape.
- If labels are written on the tape, the dataset name will be the name given in the JCL for the tape, unless EXPDT=98000 is given when a dataset name of 17 "0"s is used. Some tape management systems will not allow a tape to be used as scratch if the name in the tape labels is different from the name recorded in its database, unless the name is 17 "0"s indicating it has been initialized.

Most tape management systems provide the ability to bypass their operation on a particular TAPE DD Statement; in many cases, EXPDT=98000 in the JCL is used for this bypass (but check your tape management documentation to be sure). This may be required for certain operations with FATS. For example, if multiple volumes are being verified (read) by FATS, and their dataset names differ, FATS will open them all with the dataset name on the TAPE DD Statement; this may fail unless the tape management system is bypassed.

## 11.7 REPORT FORMATS

**DETAIL REPORT** The FATS detail report contains one line for each significant event which occurs on a tape being processed by FATS. Significant events include labels written or saved, permanent or temporary data checks, end-of-tape (TAPE INDICATE), tape marks read, and certain error conditions. Each detail line will include:

- The TAPE DD name (ID), e.g., TAPE1
- The tape unit address (UCB) as a 4-digit hex number, e.g., 0381 or 4751
- The tape volume serial (LABEL)
- The FATS operation keyword (OPTION), e.g., WRITE
- The PASS number (always 1).
- The file number (FILE NO) and record number (RECORDS) within the file (for READ only)
- The LOCATION, in feet or meters from the beginning of the tape, of the event
- The LENGTH, in inches or centimeters, of any permanent data check (if contiguous data checks occur, this length is cumulative)
- The number of RETRIES attempted for any data check
- An ACTION message detailing the type of event (these messages are listed in the “Messages and Codes” section of this manual).

From this detail report, it is possible to see where the errors are and how they are grouped. For a WRITE operation, the location listed for TAPE INDICATE (end-of-tape) shows the total length of the tape. For a READ operation, the line for TAPE MARK gives the number of records in the file and length of tape read to that point (the length is cumulative, so you must subtract to get individual file lengths).

By default, FATS will print a blank line for every four detail report lines, and will print a blank line whenever a new volume is mounted on a tape unit. If the detail report is printed on a separate report file (DD name SYSPRINx rather than SYSPRINT), options are available to control the number of detail lines grouped together, and to skip to a new page for a new volume.

**SUMMARY REPORT** The FATS summary report will contain several lines for each tape volume processed by FATS. The first line will contain ID, UCB, LABEL, and option (same as in the detail report above), plus the completion code (NORMAL or ABNORM), and the total length, in feet or meters, of the tape as processed by FATS.

The second line will appear once or twice (if two passes are done), and gives the number of data checks encountered on the tape summarized by the number of retries performed for each. Counters for retry levels 1 through 10 appear on this line; the word “\*PERM” appears above the column which is considered a permanent error by FATS (the value of the RETRY= parameter); non-zero counts for any lesser retry values indicate temporary errors. If RETRY= had a value greater than 10, this line will contain only the total number of temporary and permanent data checks, with appropriate headings above.

## 11.7 CONTINUED . . .

**FATS CONTROL REPORT**

```

● FAST ANALYSIS OF TAPE SURFACES - CONTROL - FATS VER 4.8.01 INNOVATION DATA PROCESSING AUTHORIZED 06/01/2001 PAGE 1
● FATS100 FOR INSTRUCTIONS AND MESSAGES AND CODES USE PARM=I
● FATS110 WRITE(1) MAXERR=3,
● FATS110 VOL=(001210,001540,003320)

● FATS111 CHARACTERISTICS OF THE TAPES TO BE ANALYZED
FATS111 ID PASS OPTION LABEL FILES RETRY MODE BPI GAP REWIND THRESHOLD MAXERR BLKSIZE
FATS111 TAPE1 1 WRITE SAV 10 D4 038000 3040 YES 01000 00003 65528
● FATS301 END OF REPORT

```

**FATS DETAIL REPORT**

```

● FAST ANALYSIS OF TAPE SURFACES - DETAIL - FATS VER 4.8.01 INNOVATION DATA PROCESSING AUTHORIZED 06/01/2001 PAGE 1
● MESSAGE ID UCB LABEL OPTION PASS FILE NO RECORDS LOCATION LENGTH RETRIES ACTION
● FATS107 TAPE1 0381 001210 WRITE 1 0 FT LABEL SAVED
● FATS107 TAPE1 0381 001210 WRITE 1 5 FT 05 TEMP DATA CHECK
● FATS107 TAPE1 0381 001210 WRITE 1 56 FT 05 TEMP DATA CHECK
● FATS107 TAPE1 0381 001210 WRITE 1 67 FT 08 TEMP DATA CHECK

● FATS107 TAPE1 0381 001210 WRITE 1 104 FT 1 IN 10 PERM DATA CHECK
● FATS107 TAPE1 0381 001210 WRITE 1 104 FT 2 IN 10 PERM DATA CHECK
● FATS107 TAPE1 0381 001210 WRITE 1 105 FT 3 IN 10 MAX ERROR EXCEEDED *

● FATS107 TAPE1 0381 001540 WRITE 1 0 FT LABEL SAVED
● FATS107 TAPE1 0381 001540 WRITE 1 384 FT 02 TEMP DATA CHECK
● FATS107 TAPE1 0381 001540 WRITE 1 506 FT TAPE INDICATE *****

● FATS107 TAPE1 0381 003320 WRITE 1 0 FT LABEL SAVED
● FATS107 TAPE1 0381 003320 WRITE 1 510 FT TAPE INDICATE *****
● FATS301 END OF REPORT

```

**FATS SUMMARY REPORT**

```

● FAST ANALYSIS OF TAPE SURFACES - SUMMARY - FATS VER 4.8.01 INNOVATION DATA PROCESSING AUTHORIZED 06/01/2001 PAGE 1
● MESSAGE ID UCB LABEL OPTION COMP LENGTH ***** DATA CHECKS BY NUMBER OF RETRIES *****
● CODE ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE TEN
● FATS300 TAPE1 0381 001210 WRITE ABNORM 00105 FT *PERM
● FATS300 PASS 1 0 0 0 0 2 0 0 1 0 3
● FATS300 TAPE1 0381 001540 WRITE NORMAL 00506 FT *PERM
● FATS300 PASS 1 0 1 0 0 0 0 0 0 0 0
● FATS300 TAPE1 0381 003320 WRITE NORMAL 00510 FT *PERM
● FATS300 PASS 1 NO DATA CHECKS ON THIS TAPE
● FATS301 END OF REPORT

```

## 11.8 SECURITY

**STANDARD  
FATS SECURITY**

FATS will OPEN the input and output tapes, so your security system will be invoked. If you have configured it to protect tape datasets and/or tape volumes, those security rules will apply, so the userid under which FATS is run must be authorized to read the input volumes and datasets and write to the output volumes and to create the output dataset names used. However, see the notes on BLP below.

**PROTECTING  
FATS  
FUNCTIONS**

FATS allows you to use your security system to control which users are authorized to execute each of the FATS functions. For example, you might authorize all operators to LABEL tapes, but restrict the ability to ERASE a tape to supervisors. **These security checks are disabled by default; see Section 90 for instructions on enabling them.**

Use of the FATS functions is controlled by resources defined in the RACF FACILITY class (or its equivalent in other security systems). Consult your IBM RACF or other security vendor documentation for the procedure for defining such FACILITY class resources. Once defined, you must grant READ authority to the resource for users authorized to use it.

For compatibility with earlier releases of FATS, the default is to allow all users to execute all functions. If your security system indicates to FATS that a given resource name is not defined, all users will be able to use that function. If you define some but not all of the resources, only those functions are protected. For example, if you don't define FATS.READ, all users can execute FATS READ functions.

The five FACILITY class resources checked by FATS are:

FATS.READ	use the READ (verify) function
FATS.WRITE	use the WRITE (certify) function
FATS.LABEL	use the LABEL (label tapes) function
FATS.ERASE	use the ERASE (erase tapes) function
FATS.BLP	use the BLP operand or LABEL=(,BLP) in JCL.

As each control statement is processed, FATS will issue a RACROUTE to verify that the user has at least READ authority to the proper resource name. The step will fail if RACROUTE returns a code of 8 or higher, meaning that the resource is defined to your security system but the user associated with the FATS step does not have authority to it.

There is no FACILITY resource for the ANALYZE function. ANALYZE invokes FATAR, which has its own set of security checks as defined in Section 21.6.

**BYPASS LABEL  
PROCESSING  
(BLP)**

When label processing on input or output volumes is bypassed using the BLP option in JCL or the BLP operand on the FATS statements, volume labels and dataset names are not verified by OPEN and so authorization to the volume and datasets cannot be guaranteed; if anything, your security system will check authority to the volser and dsname provided in the JCL when BLP is used, these may not match the actual tape. For this reason, most security systems allow your installation to restrict the use of BLP.

However, some functions of FATS (such as labeling or certifying virgin tapes) **require** the use of BLP. Innovation suggests that you provide BLP authority to a limited number of users (operators, tape librarians, and/or system programmers) so that they can run these functions when required.

Resource FATS.BLP controls the ability to use the BLP operand on FATS control statements and the ability to put LABEL=(,BLP) on a TAPEX DD statement. When opening a tape, FATS will check if BLP is being used, from either source, and will fail the step if the resource is defined but the user is not authorized. If the user is authorized to FATS.BLP but you have a general control on the use of BLP defined in your security system, the user must also be authorized to that resource.

**12.0 FATS GUIDELINES****USAGE  
GUIDELINES**

Many customers have asked for suggestions on the use of FATS as part of a regular program of tape maintenance. The following section contains explanations and guidelines designed to help you in the development of such a program. However, each installation should develop its own cost-justified procedure establishing what level of certification is required, at what point tapes are to be discarded, how often archival tapes are to be verified, and so forth. These guidelines are presented only as an example and an aid.

This section also contains answers to questions on FATS which customers have often asked.

**12.1 CAUSES OF TAPE ERRORS****CAUSES OF  
TAPE ERRORS**

To better understand the usage of FATS you must understand the causes of error on magnetic tape. Any of the following can cause errors:

- 1) Excessive oxide on the tape surface.
- 2) Dirt or debris on the surface (often microscopic).
- 3) An oxide coating on the tape surface which does not retain an acceptable signal.
- 4) Tape surface damage such as pinholes, scratches, creases, and edge damage.
- 5) Dirt or oxide buildup on the tape drive heads.
- 6) Malfunction of the tape drive itself.

Many errors may be caused by loose oxide or dirt on the tape surface. These errors may show up as temporary (meaning that FATS retry actions caused the contaminant to be dislodged), or permanent. Even if they are permanent, surface cleaning by a stand-alone tape cleaning machine, or repeated certifications by FATS, may cause the errors to disappear.

Truly permanent errors may be caused by defective oxide coatings or surface damage. These errors cannot be rectified by FATS or a tape cleaner.

Dirty tape heads or tape drive malfunctions may cause reported errors which do not truly exist on the tape itself. FATS depends on the proper functioning of the tape drive to detect and report errors, so any errors in the tape drive mechanism will produce spurious errors. We recommend that the tape drive heads be cleaned at regular intervals when certifying tapes (e.g., every 10 or 20 tapes, more often if certifying brand new tapes); the FATS parameters MAXCLEAN= and ERRCLEAN= can be used to insure this. If FATS reports many errors on a tape, we recommend re-certifying it on another tape drive before deciding to discard it.



## 12.2 OPERATING SYSTEM VS. FATS ERRORS

### OPERATING SYSTEM VS. FATS ERRORS

We are often asked why a tape on which FATS has reported one or more errors can be used successfully as an output tape by an application program. The answer has to do with the purpose of FATS error reporting versus the operating system error recovery (full details of error recovery and FATS operation are found in Section 11 "FATS Technical Summary").

### FATS PURPOSE

FATS certification has the purpose of identifying every section of the tape (in increments of a few inches) which has a permanent or temporary write error. As explained below, this can allow you to predict where the operating system will experience permanent write errors or degradation due to error recovery.

### OPERATING SYSTEM ERROR RECOVERY

Operating system error recovery has the purpose of letting the application program run to successful completion despite I/O errors on tape. To accomplish this, write errors are retried by erasing a length of tape and attempting to write the block again. Depending on the location of the error, and the length of the block, this action may need to be repeated until it is successful or until error recovery considers it a permanent error at which point the application program is usually terminated.

As an example, assume that a tape has a permanent error in the middle of a tape with no other errors around it. That tape is being used by an application program writing blocks of length 32000 bytes (about 1 inch on a 3480/3490 cartridge). At the first attempt to write a block, that permanent error falls in the last fraction of the block. Hardware error recovery erases about .5 inches of tape and writes again, but the error is still within the block (now about in the middle). The system erases .5 inches again, but the error occurs again, right near the beginning of the block. Another .5 inches are erased again, and the block is now written past the error and writes successfully. No error is apparent to the program or the operator, but time was wasted doing this recovery, and 1.5 inches of blank tape were wasted.

If however, there is a cluster of permanent errors on the tape, spaced about an inch apart, this same situation will probably result in a permanent write error indication from error recovery. As error recovery erases tape (.5 inches at a time) to move the block up, it keeps encountering a new error on the tape, since the data block is longer than the distance between errors. Obviously, a given pattern of permanent errors reported by FATS may or may not result in a permanent error to an application program depending on the tape density and data block length.

### TAPE ERROR STATISTICS

Although tape write errors may be bypassed by this error recovery process, the existence of the errors is always recorded by the operating system. The system error recording file "SYS1.LOGREC" will always contain the total number of temporary and permanent read and write errors encountered on every tape volume processed, and can be printed by the EREP (Environmental Recording, Editing, and Printing) program appropriate for your system. If the SMF sub-system is set up to record type 21 ESV (Error Statistics by Volume), these records will also contain this information and can be printed by the IFHSTATR utility documented in the appropriate system utilities manual. These statistics may help to make clear the hidden cost of "unseen" tape errors.

**12.3 VARIATIONS IN FATS RESULTS**

We are often asked why the results of a FATS certification or verification will vary on two consecutive runs on the same tape. If you review the causes of tape errors above, the reasons are fairly obvious.

For errors caused by dirt or debris on the tape, these contaminants may be removed by one FATS execution, or may be picked up by the tape drive heads and re-deposited in another location, causing an error to “move”, even if re-certified on the same drive.

If errors are caused by marginal oxide surface on the tape, the surface may be able to record and reread a signal using the amplifiers and heads on one tape drive, while being just barely inadequate on another, causing errors to appear and disappear.

Naturally, if the errors are caused by dirty tape heads or tape drive malfunction, the results will be different on different drives.

## 12.4 ANALYSIS OF CERTIFICATION RESULTS

FATS certification (WRITE function) is designed for new tapes or tapes currently in scratch status in order to decide if they are acceptable for use as output tapes. It is impossible to make firm recommendations on the action to take on a particular tape based on its FATS results; each installation must decide for itself what level of error recovery degradation it is willing to accept (based on the number of permanent errors on a tape) versus the cost of replacing a tape volume. However, we can provide some guidelines for arriving at this value, as well as some circumstances when action is definitely indicated.

These decisions will usually require examination of the FATS detail report to analyze the number and position of the errors on a given tape. It is possible to automate some of this decision process (the FATS parameters MAXERR= and MAXCERR= will cause automatic rejection of a tape with an excessive number of errors or contiguous errors), but most of the analysis is beyond the capabilities of FATS to perform dynamically. The user will have to examine the total number of permanent errors, the location on the tape of these errors, and the clustering of the errors (FATS will report on errors which are contiguous, but some errors may be separated by small lengths of good tape but still be close enough to cause problems).

In most installations, the majority of tape datasets occupy only the first 10-20% of the tape volume. As a result, the beginning of the tape gets the most repeated use and is subject to the most errors due to tape wear. If a tape shows permanent errors (even only 1 or 2) in the early part of the tape, some action is indicated. It is possible to cut off the portion of the tape containing the errors (except tape cartridges) and put a new beginning-of-tape reflective marker on it, resulting in a shorter but usable tape. Most installations set some minimum acceptable length for a tape (such as 2000 or 2200 feet); since FATS shows both the total length of the tape and the length to remove (the location of the last error), you can easily calculate whether a tape is worth this "stripping" action. For cartridge tapes, it is possible on 3480/3490/3490E cartridges to remove a small part of the tape and connect a new leader block, but the tape drives will reject tapes that are very much shorter than the standard lengths, so this is usually not practical.

Next, look for contiguous or clustered permanent errors. Clusters of errors exceeding a few inches in length on cartridges or 3 feet in length on round tapes are almost certain to cause permanent write errors; clusters shorter than this may be recoverable but will cause degradation. If a tape contains any long clusters or more than one short cluster, it should be removed from use.

If a tape is free from the above problems, it can be categorized by the number of permanent errors. The following categories are suggestions only:

### For Round Tape:

- |                |  |
|----------------|--|
| 0-10 Errors    | – Good tape  |
| 11-25 Errors   | – Marginal tape. Clean if possible. Do not use for critical applications.          |
| Over 25 Errors | – Bad tape. If possible, clean and re-certify. If errors persist, remove from use. |

### For Cartridges:

- |               |   |
|---------------|---|
| 0-5 Errors    | – Good tape   |
| Over 5 Errors | – Clean and re-certify. If errors persist, remove from use. |

**12.5 ANALYSIS OF VERIFICATION RESULTS****VERIFICATION  
RESULTS**

FATS verification (READ or ANALYZE functions) have the purpose of verifying the readability of existing data files on tape. This can be used for various critical production data (to avoid data checks in the middle of critical runs) or verification of archival data which is rarely read but which must be retained.

FATS will report on every data block (by file number and block number within file) on which a permanent or temporary data check occurred. FATS does not attempt to reproduce the complete actions of the operating system error recovery in the case of a read data check, so it is possible that a data check reported by FATS will in fact be readable by an application program.

However, even one data check on such a tape means that the tape is suspect and may cause application failures now or in the future. To be safe, the tape should be copied as soon as possible to a certified scratch tape. Because of its superior facilities for recovering from data checks and its ability to copy multiple files, the FATAR program is recommended for this purpose.

**12.6 RECOMMENDATIONS FOR FATS USAGE**

The following is a suggestion for a formal program of FATS usage to certify scratch tapes and verify data files. As explained above, each installation will have to select the intervals and values that make sense for it.

**ARCHIVAL  
DATA TAPES**

Archival data tapes (long term retention tapes which are rarely read but must be retained for legal or other reasons) should be verified for readability at intervals not exceeding two years. Tapes which sit on the rack for long periods have a tendency to "dry out" causing data checks. The action of reading them with FATS will help to prevent this as well as identifying any data checks which have occurred. Any data check is good reason to copy the tape.

**CRITICAL  
DATA TAPES**

It is reasonable to assume that a tape which was recently written successfully will be readable without error, but if certain data tapes are critical to the successful execution of important programs, especially if those programs are long-running or difficult to restart, it is useful to run FATS against each data file before starting the application. This verification only takes a few minutes per tape, but may save hours of recovery if data checks are found.

**NEW TAPES**

New tapes should be cleaned (if a manual cleaner is available), and certified and labeled using FATS. If the tapes cannot be separately cleaned and they do get errors, certify them several times to remove all loose oxide; be sure to clean the tape drives frequently. If certification of a new tape discovers errors which would classify it as a marginal or bad tape, you may wish to discuss it with your tape supplier.

**SCRATCH  
TAPES**

Tapes which are in scratch status (available for use as output tapes), should be certified if:

- a) A certain interval (such as 4 months) has elapsed since their last cleaning
- b) They have been used for output more than a certain number of times (such as 50)
- c) They have received more than a certain number of temporary or permanent read or write errors (such as 5) as reported by SMF or EREP.

Some tape management systems will track these numbers and recommend cleaning/certification of specific volumes based on user provided thresholds. Disposition of the certified tape should be based on the recommended analysis technique outlined above.

## 12.7 QUESTIONS AND ANSWERS

These are some commonly asked questions about FATS.

### DATA CHECKS AT EOY

#### Why do round tapes sometimes show data checks at end of reel?

The end of a round tape reel is indicated by a reflective marker on the tape. If the tape is wound too tightly by the manufacturer or by the tape drive, impressions of this marker will appear every 14 inches for several layers of tape above and below the reflector. These impressions can cause data checks, especially when writing large data blocks. FATS will report these errors as data checks about 1 foot apart at the end of the tape. The end of tape reflector can be moved up to eliminate this error.

Cartridge tapes do not use physical markers for beginning and end of tape and are not subject to this kind of error.

### FATS RESOURCES

#### What resources does FATS use?

FATS CPU usage is minimal, but, by the nature of the function it performs, it is a heavy user of the tape channels and control units. We recommend that FATS be run primarily when other tape usage is low. If multiple FATS operations are being performed, FATS will perform better if the tape drives are on separate channels and control units, if possible. There is little point in allocating more than three tape drives on the same channel or control unit to FATS, since no performance gain will result. Note that this does not apply to the FATS ERASE function which does not tie up the channel or control unit. Also, tape cartridge sub-systems, because of their multiple channel paths and buffering, may be able to achieve better performance under high load.

### SMF/EREP RECORDING

#### Why are FATS data check reports different from what I see in SMF/EREP?

Temporary and permanent data checks detected by FATS may not be reported differently (or not reported at all) in SMF (type 21 ESV records) and "SYS1.LOGREC" (EREP). FATS is able to do what it does only by inhibiting system and hardware error recovery. This causes errors on round tape to go unrecorded, while errors on cartridges are reported; however, every FATS retry will count as one error, so the SMF/EREP values will be much higher than the FATS totals. Operators and hardware maintenance personnel should be instructed to ignore errors recorded because of FATS runs, since they do not indicate the type of device or volume problems that usually invoke such records.

### RETRY LEVEL

#### What retry level should be used?

Although the default value for RETRY= in FATS is 10, experience and analysis of many FATS reports has shown that if an error repeats 5 times, it is likely to still be present after 10 retries. If a tape has many errors, specifying RETRY=5 may significantly reduce the elapsed time of the certification, while still providing the same information on the condition of the tape.

### FULL-TAPE VERIFY

#### Why can't FATS READ verify to the physical end-of-tape?

Many customers have asked for the ability in FATS to READ verify to the physical end-of-tape reflector, rather than stopping at two tape marks (logical end-of-data). This is not practical for several reasons:

- a) The tape drive raises an exception condition when you attempt to write past the end-of-tape (this is how the FATS WRITE operation knows when to stop), but it gives no indication when you are reading past the physical end-of-tape. It would be likely that FATS would read past EOT, which can cause errors and (on round tapes) require operator intervention to rethread the tape.
- b) It is possible to discover if the tape is positioned past the reflector (at the cost of an extra I/O operation for every data block), but this will work only if there are data blocks recorded out to the end-of-tape. If there is blank tape preceding the reflector, the first read in this blank area will not end until the tape runs off the reel; FATS has no way to predict and avoid this.

In any case, the function of FATS verification is to prove the readability of the recorded data on the tape. There is no point in verifying data past the current logical end-of-data.

## 13.0 FATS EXECUTION JCL

To execute FATS, the following JCL statements are required:

**EXEC  
STATEMENT**

The EXEC statement specifies the FATS program name (PGM=FATS), memory requirements (REGION=, if your installation defaults are insufficient), and parameter field (PARM=).

The basic memory requirements of FATS are 256K. If FATS' companion product FATAR is invoked as a FATS sub-task (via the ANALYZE(n) control statement), then the memory requirements of each concurrent FATAR sub-task must be added to the basic requirement (FATAR requirements are documented in Section 22 of this manual).

If PARM=I is specified on the EXEC statement, FATS will print on SYSPRINT a brief summary of its control statements and messages.

**STEPLIB/JOBLIB  
DD STATEMENT**

A STEPLIB or JOBLIB DD Statement will be required if FATS has been linked into a private library. It can be omitted if FATS is in a system library which can be accessed without a STEPLIB/JOBLIB statement (that is, a library in the system link list). The library must be an MVS APF authorized library.

**SYSPRINT DD  
STATEMENT**

SYSPRINT receives the listing of all FATS control statements, a summary of the operations to be performed, and messages about major errors. In addition, it may receive the detail reports for each tape drive and the summary report if the DD Statements for those reports are absent. SYSPRINT is normally allocated to a SYSOUT dataset. Its DCB attributes are RECFM=FBA, LRECL=121. If blocksize is specified, it must be a multiple of 121, otherwise it will default to 121 for SYSOUT or 1210 for other devices.

**SYSPRINx DD  
STATEMENT**

SYSPRINx receives the detail report for the operations performed against tapes mounted on DD Statement TAPEx, where "x" is a digit from 1 to 9. This allows the detail reports for simultaneous FATS operations to be printed separately. SYSPRINx is normally allocated to a SYSOUT dataset. If SYSPRINx is not present for a particular TAPEx, its detail report will go to SYSPRINT; however, SYSPRINx DD statements are required when running ANALYZE operations (FATAR under FATS). DCB attributes for SYSPRINx are the same as those for SYSPRINT.

**SUMMPRT DD  
STATEMENT**

SUMMPRT receives the summary report, consisting of several lines summarizing the results of FATS operation against each tape processed. SUMMPRT is normally allocated to a SYSOUT dataset. If SUMMPRT is not present, summary reports will go to the SYSPRINx dataset associated with the TAPEx DD Statement on which the tape was mounted; if it is also absent, summary reports go to SYSPRINT. DCB attributes are the same as for SYSPRINT.

**TAPESUMM DD  
STATEMENT**

When running ANALYZE operations (FATAR under FATS), TAPESUMM may optionally be included to receive the FATAR tape summaries, including dataset name, tape label information, block and byte counts, lengths, minimum/maximum/average blocksizes and error counts. DCB attributes are the same as for SYSPRINT.

**SYSUDUMP DD  
STATEMENT**

SYSUDUMP requests an abend dump if major errors occur (note that most internal abends in FATS are for the user's information only and do not cause dumps). SYSUDUMP is usually allocated to SYSOUT. If you have the ABEND-AID product from COMPUWARE include the following so that a fully-formatted dump is produced:

```
//ABNLIGNR DD DUMMY
```

**TAPEx DD  
STATEMENT**

TAPEx defines a tape unit (and optionally a volume) to be used for one operation by FATS. "x" is a digit from 1 to 9 and corresponds to the digit on a FATS control statement (e.g., WRITE(1) uses DD statement TAPE1).

The DD statement must specify a UNIT= parameter appropriate to allocate the type of tape drive required for the tapes to be processed. This can be a generic device type (such as UNIT=3480, UNIT=3590-1) or an installation-defined esoteric name (such as UNIT=TAPE, UNIT=CART).

## 13.0 CONTINUED . . .

**TAPEX DD  
STATEMENT  
(continued)**

Any type of label processing may be specified (SL, AL, NL, BLP), but it is frequently desirable with FATS to specify LABEL=(,BLP). If installation conventions do not allow the use of BLP, the FATS parameter BLP can be used to internally change the label type to BLP before open; see Section 11.8 for security considerations for the use of BLP.

**WARNING:** For proper operation of FATS, the TAPEX DD statement must contain a DSN= parameter and it must not specify a temporary (&&) name. Also the parameters DISP=(...,PASS) or VOL=(...,RETAIN) must not be specified. Otherwise, the tape may forward space to the end of tape or other errors may occur.

For output volumes (WRITE, LABEL, ERASE):

The TAPEX DD statement will usually say DISP=(NEW,KEEP). A volume serial (VOL=SER=) can be specified to request a specific volume, or omitted to request a scratch volume allowing the operator to mount any appropriate volume.

The volume serial can also be omitted when FATS is supplying the serials internally (via its own VOL= parameter). In this case, the JCL parameter DEFER (as in UNIT=(xxxx,,DEFER)) should be used so that the operating system will not call for a tape mount before FATS fills in the proper volume serial.

**NOTE:** if a volume serial is specified, the label type is SL, and the FATS MULT parameter is given, for subsequent mounts (after the first) the volume serial will be blanked to request scratch tapes; if this wasn't done, MVS would just reuse the tape already mounted and process it over and over.

Examples:

```
//TAPE1 DD DSN=FATS.OUT1,UNIT=3490,
//      DISP=(,KEEP),LABEL=(,BLP)
//TAPE2 DD DSN=DUMMY,UNIT=(TAPE,,DEFER),
//      DISP=(NEW,KEEP)
```

For input volumes (READ, ANALYZE):

DISP=(OLD,KEEP) should be specified, and a dataset name (DSN=), unit (UNIT=) and volume serial (VOL=SER=) must be given (if the FATS parameter VOL= is being used to supply serial numbers, the volume serial in JCL may be any dummy serial, e.g., VOL=SER=DUMMY). If LABEL=(,BLP) is specified, the dataset name and volume serial do not have to be the actual values.

If the FATS parameter VOL= is used to supply volume serials, the JCL parameter DEFER (as in UNIT=(xxxx,,DEFER)) should be used so that the operating system will not call for a tape mount before FATS fills in the proper volume serial.

Examples:

```
//TAPE1 DD DSN=INPUT.DSN,UNIT=3490
//      VOL=SER=901234,DISP=(OLD,KEEP)
//TAPE2 DD DSN=DUMMY,UNIT=(TAPE,,DEFER),
//      VOL=SER=DUMMY,DISP=OLD,LABEL=(,BLP)
```

**TAPEXOUT DD  
STATEMENT**

TAPEXOUT (where "x" is the same 1 digit number as in TAPEX above) is used only for the FATS ANALYZE(x) function, which causes the companion product FATAR to be invoked. If present, TAPEXOUT will request FATAR to copy the tape mounted on TAPEX. See FATAR documentation elsewhere in this manual for details.

**SYSIN DD  
STATEMENT**

The SYSIN DD Statement is the source of FATS control statements. It is normally a "DD \*" spool file, but can be any disk or tape file with DCB characteristics RECFM=FB and LRECL=80.



### 13.1 OPERATOR CONTROL

**OPERATOR CONTROL** FATS will terminate by itself when all requested operations have completed. However, sometimes you may wish to terminate FATS early (especially when processing a long list of tape volumes) or to terminate processing of a particular tape volume. There are 2 ways to do this:

- by default, FATS issues a WTOR to the operator's console (message FATSW02). This message can be left outstanding and will delete itself when FATS terminates. However, you may reply to it at any time to terminate FATS or terminate processing of a tape volume. The WTOR can be suppressed by the WTOR=NO operand.
- if the optional MODIFY=YES parameter is specified, MVS console command STOP (P) and MODIFY (F) can be used for the same purposes.

**WTOR RESPONSES** The FATSW02 message can be replied to at any time with one of these replies ("nn" is the WTOR reply number associated with the message):

- R nn,EOJ** – FATS will complete processing of all tapes currently in progress, then will terminate. If any tape mounts are pending, they must be satisfied (those tapes will be processed before termination).
- R nn,KEOJ** – FATS will terminate immediately with a U0888 abend or return code 12. Tapes in progress will be immediately halted.
- R nn,Kuuuu** – "uuuu" is the MVS 3- or 4-digit hexadecimal device address of a tape drive currently in use by FATS. The tape volume being processed on that device will be immediately halted; however, another tape may be mounted on that drive if called for by the FATS control statements. Other drives in use by FATS will not be affected.

**STOP/MODIFY COMMANDS** If MODIFY=YES was specified, the MVS console commands STOP (abbreviated P) and MODIFY (abbreviated F) can be used for the same purposes. In the examples below, "id" is the jobname of the FATS job or started task:

- P id** – stops FATS immediately and is equivalent to the KEOJ reply above.
- F id,QUIT** – terminates FATS after all tapes which are currently in progress have completed and is equivalent to the EOJ reply above.
- F id,Kuuuu** – stops processing on the specified tape unit (3- or 4-digit hexadecimal device address "uuuu") and is equivalent to the Kuuuu reply above.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

**14.0 FATS CONTROL STATEMENTS****14.1 GENERAL****GENERAL  
RULES**

All FATS control statements must be input on 80-character records of which only columns 1 to 71 are used by FATS. Each control statement contains:

**OPERATION  
KEYWORD**

Identifies the operation to be performed by FATS. It can start in any column from 1 to 50 on the input record, but any columns before it must be blank. Only one operation keyword can be specified for each tape drive used by FATS. Valid operation keywords are:

- DEFAULT     – Set defaults for other operations
- WRITE       – Certify a tape
- READ        – Verify readability of a tape
- ERASE       – Erase data from a tape
- LABEL       – Write standard labels on a tape
- ANALYZE     – Invoke FATAR to read/copy a tape

**TAPE NUMBER**

Immediately follows the operation keyword (no intervening blank spaces) and is enclosed in parentheses. It must be a single digit from 1 to 9 and identifies to which TAPE<sub>x</sub> DD Statement this operation applies (e.g., "(1)" indicates DD Statement TAPE1). Each digit may appear on only one control statement in a given FATS execution. The tape number is not used on the DEFAULT control statement, but is required on all other control statements.

**OPERANDS**

Are separated from the operation keyword (and tape number) by one or more blank spaces. Each control statement may have one or more operands (separated by commas) or none at all. The operands modify the function of the operation keyword; they control labeling, tape mode, error retries, etc. The last operand must be followed by a blank space.

**CONTINUATION**

Operands can be continued onto another control statement by following the last operand with a comma and a blank space, and continuing the operands in any column of the next statement.

**COMMENTS**

May appear on any FATS control statement. They must be separated from the last operand by at least one blank space (a control statement containing no options cannot contain comments). Any control statement with an asterisk ("\*") in column 1 will be bypassed by FATS and treated entirely as comments.

**14.2 OPERATION KEYWORDS**

- DEFAULT** Used to change the defaults of FATS operands. It affects any other operation keywords which follow it. If several operations are to be used in a FATS step, and they all require one or more of the same operand specifications, DEFAULT is a simple way to enter the options only once. Operands specified on DEFAULT can still be overridden on a specific operation statement.
- WRITE(n)** Used to certify the quality of new or old scratch tapes by writing special patterns on the entire length of the tape and checking for write errors. **This will destroy any existing data on the tape, so the tape must be in scratch status.**
- READ(n)** Used to verify the readability of existing data on a tape. READ(n) can read one or more data files, but cannot read past the end of currently recorded data since FATS has no way of detecting the end-of-tape marker in read mode. Can be used on non-scratch tapes since it will not destroy any data.
- ERASE(n)** Invokes the "Data Security Erase" hardware function to **erase all data** from a tape. It may be used to totally erase a tape, or to erase data past currently existing files. Unlike WRITE(n), the "Data Security Erase" does not involve the tape control unit, so other FATS operations or tape I/O from other jobs can proceed without contention.
- WARNING: ERASE will erase all data from a tape or cartridge with no possibility of recovery. Use with caution.***
- LABEL(n)** Writes an IBM standard volume label and dataset labels on a scratch tape. LABEL(n) can be used to label new tapes or to re-initialize old tapes. **It does destroy any data on the tape.** Can also create unlabeled tapes.
- ANALYZE(n)** Invokes FATS' companion product FATAR to read and optionally copy a tape. Complete information on FATAR is contained in other sections of this manual. Although FATAR can be executed as an independent program, execution under FATS allows multiple FATAR tasks to execute in one jobstep and allows the use of FATS parameters for the selection of tapes to be processed. Your installation must be licensed for FATAR to use this operation.

### 14.3 OPERANDS

All operands can appear on all operation statements, but some actually apply only to one or more of the operation keywords. They are grouped together by their functional type (as shown in the section headings below), but parameters which apply only to one or more operations are marked as such. All option parameters (except VOL=) can appear on a DEFAULT Statement.

#### 14.3.1 RETRY/ERROR OPERANDS

**ERRCLEAN=n** Applies to: WRITE, READ

If more than "n" permanent or temporary errors are encountered on a tape, FATS will request the operator to clean the tape drive, and then re-certify or re-verify that tape. "n" may be from 1 to 32767. This cleaning action will be requested only once per tape drive.

Default: 32767

**MAXCERR=n** Applies to: WRITE, READ

If contiguous permanent errors totaling more than "n" inches are encountered on a tape, FATS will terminate processing of that tape. "n" may be from 1 to 32767.

Default: 200

**MAXERR=n** Applies to: WRITE, READ, ANALYZE

If more than "n" permanent errors are encountered on a tape, FATS will stop processing of that tape. "n" may be from 1 to 32767.

Default: 200

**MAXTERR=n** Applies to: WRITE, READ, ANALYZE

If more than "n" temporary errors are encountered on a tape, FATS will stop processing of that tape. "n" may be from 1 to 32767.

Default: 200

**RETRY=h** Applies to: WRITE, READ, ANALYZE

**RETRY=(l,h)**

"h" defines number of retries required before an error is considered permanent, and "l" the number of retries before an error (temporary or permanent) will be printed in the detail report. "h" may have values from 1 to 99, and "l" may go from 0 to 99 but must be less than "h". The value of "h" affects the summary report: for 10 or less the summary report will show temporary and permanent errors by number of retries; for over 10 all temporary and permanent errors will be totaled.

Default: (1,10)

**STOP** Applies to: WRITE, READ

When this option is specified, FATS will stop and issue message FATSW05 to the system operator whenever a permanent error is encountered, allowing inspection of the tape. After inspection, the operator may reply to the message requesting continuation (and stopping at the next error), termination, or continuation without stopping. Physical inspection of the tape may not be possible on all types of tape drives.

Default: Do not stop on errors.

**STOPNUM=n** Applies to: WRITE, READ

Invokes the STOP option (above) after "n" permanent errors have been encountered, allowing the first "n" errors to be skipped without operator interaction. Do not specify STOP if STOPNUM=n is specified.

Default: Do not stop on errors.

### 14.3 CONTINUED . . .

#### 14.3.2 VOLUME MOUNTING OPERANDS

**BLP** Applies to: ALL

The BLP operand is provided to help when installation conventions do not allow LABEL=(,BLP) to be specified on a JCL DD statement, but it desired to bypass label processing on FATS tapes. BLP processing may be required when certifying/labeling virgin or unknown tapes, or when reading multiple input tapes with varying dataset names. When BLP is specified, the JCL label parameter is internally changed to BLP before the tape is opened. See Section 11.8 for security considerations for the use of BLP.

Default: Honor the JCL LABEL= parameter.

**MAXCLEAN=n** Applies to: ALL

When multiple tapes are being processed on a tape drive, specifies that the system operator is to be requested (via console message FATSW06) to clean the tape drive after every "n" tapes are completed. This is to prevent dirt or oxide deposited on the read/write heads by previous tapes from causing spurious errors.

Default: No cleaning requested between tapes.

**MAXVOLN=n** Applies to: ALL

Limits the number of volumes which FATS will request on a given tape drive when either the MULT or VOLINCR=n operands are used.

Default: No limit for MULT or 682 for VOLINCR=n.

**MULT**  
**NOMULT** Applies to: ALL

MULT requests that multiple tapes be processed. As each tape is completed, a mount for a new tape will be issued. This will continue until the operator terminates FATS (see Section 13.1), or until the number of tapes specified by the MAXVOLN=n parameter have been processed. NOMULT (the default) is used to turn off MULT if it has been specified on a DEFAULT statement.

Default: Process only one tape unless VOL= specified.

**VALIDATE=** Applies to: WRITE, ERASE, READ, ANALYZE

If label processing is bypassed because of the FATS BLP operand or LABEL=(,BLP) on TAPEX DD statements, data management will not verify the volume serial or expiration date of the tape, so it is possible that the wrong tape may be accidentally mounted and overwritten. However, if VALIDATE= is specified, FATS will verify the volume serials of the mounted tapes for input tapes (VALIDATE=INPUT), output tapes (VALIDATE=OUTPUT), or both (VALIDATE=ALL). The volume serial compared is the serial in the TAPEX DD statement or that provided by the FATS VOL= parameter. For output tapes, the expiration date in the tape labels will also be checked. If the wrong volume serial is mounted, or if the expiration date is not yet reached, FATS will issue message FATSW08 to the system operator giving the option of ignoring the error, mounting another tape, or skipping the tape. For WRITE and ERASE operations, validation will be done only if the SAVLAB operand is specified or defaulted. VALIDATE= will be ignored if the NOREWIND operand is given.

Default: No validation.

## 14.3 CONTINUED . . .

- VOL=v** Applies to: ALL (cannot appear on a DEFAULT statement).
- VOL=(v,...,v)** Specifies one or more tape volume serials to be processed by the operation statement on which it appears. "v" is a 1-to-6 character volume serial. VOL=v requests one volume. VOL=(v,v,...,v) requests multiple volumes; in this format, up to 682 serials may be supplied; the serial list may be continued onto multiple input statements by placing a blank space after any comma in the volume list, and continuing the list in any column of the next input record. When used in conjunction with the VOLINCR=n and MAXVOLN=n parameters, VOL= can be used to request a sequence of tapes without actually entering all of their serials. Note that VOL= on a LABEL statement implies that FATS will label the tapes with the given volume serial; on a WRITE or ERASE statement, this is true only if the NOSAVE parameter is also specified.
- VOLINCR=n** Applies to: ALL
- This operand can be used only in conjunction with the VOL= operand. When specified, the last (or only) volume serial specified by VOL= will be incremented by "n" which must be a 1-to-6 digit decimal number. That volume serial must end in at least one numeric digit; only the trailing numeric part of the serial is incremented. The serial will be repeatedly incremented by "n" until either 682 volume serials have been generated or until the serial cannot be incremented further without "overflowing" the numeric portion. Although a large number of such serials may be generated by VOLINCR=, the MAXVOLN=n operand may be used to limit the number of volumes which will actually be requested by FATS. VOLINCR=1 may be used to cause a number of consecutively numbered tapes to be requested.
- Default: No incremented serials are generated.

## 14.3 CONTINUED . . .

## 14.3.3 LABELING OPERANDS

**ANSI** Applies to: WRITE, LABEL, ERASE

Requests that tapes are to be labeled with ANSI V3 labels instead of IBM standard labels.

Default: IBM standard labels will be written

**DD=LABEL** Applies to: WRITE, LABEL, ERASE

Causes the tape to be labeled with the volume serial specified on the associated TAPEX DD statement. Cannot be used with the MULT parameter.

**LABEL=vvvvvv** Applies to: WRITE, LABEL, ERASE

Causes the tape to be labeled with volume serial "vvvvvv". Cannot be used with the MULT parameter.

**NOLABEL** Applies to: WRITE, ERASE, LABEL

Resets all LABEL options (including SAVLAB). The tape being processed will become an unlabeled tape. When used with the LABEL(n) statement, NOLABEL will quickly initialize unlabeled tapes.

**OPERATOR  
NOOPERATOR** Applies to: WRITE, LABEL, ERASE

If OPERATOR is specified, as each tape is opened, the system operator will receive message FATSW01 and must reply with the volume serial to be written on the tape. NOOPERATOR is used only in conjunction with SAVLAB (specified or by default) to prevent SAVLAB from going into OPERATOR mode if a tape is not labeled (see below), so that unlabeled tapes will remain unlabeled.

**OWNERID='....'** Applies to: WRITE, LABEL, ERASE

Specifies the contents of the owner ID field in volume labels written by FATS. 1 to 10 characters (including blanks) may appear between the apostrophes. OWNERID= has an effect only if FATS is writing a volume label due to other labeling options. Even if SAVLAB is used, OWNERID= may be used to change the owner ID.

Default: 10 blank spaces (or original value if SAVLAB)

**SAVLAB  
NOSAVE** Applies to: WRITE, ERASE

If SAVLAB is specified or defaulted, and the tape being certified contains IBM standard labels, the tape will be re-labeled with its original volume serial; if the tape does not contain IBM standard labels, the tape will be labeled with a serial provided by the system operator (see OPERATOR above) unless the NOOPERATOR parameter is present. NOSAVE will prevent the volume serial from being preserved. Note that SAVLAB preserves only the volume serial and owner ID from the volume label; the contents of dataset header labels is not saved.

Default: SAVLAB



**14.3 CONTINUED . . .****14.3.4 PRINT OPERANDS**

**LINECNT=n** Applies to: ALL

Specifies the number of lines per page to be printed on FATS reports. If specified on a DEFAULT statement, it will control the page size of the FATS control report and summary report as well as providing a default for the detail reports. If specified on any other operation statement, it modifies only the detail report page size for that operation.

Default: 60 lines per page

**MAXDETCNT=n** Applies to: ALL (except ANALYZE)

Specifies the maximum number of detail report lines which will be printed together before double spacing. This will be effective only if a separate detail report DD statement SYSPRINx has been provided.

Default: 4

**NEWPAGE** Applies to: ALL (except ANALYZE)

When multiple tapes are processed on a tape drive, the NEWPAGE operand requests that the detail report will skip to a new page whenever a new volume is processed, making the report easier to separate by volumes. This will be effective only if a separate detail report DD statement SYSPRINx has been provided.

Default: Skip a single blank line between volumes.

**NONMETRIC  
METRIC** Applies to: ALL

When NONMETRIC is specified (or defaulted) FATS will list tape lengths and error positions and lengths in American units (feet and inches). When METRIC is specified FATS will list tape lengths and error positions and lengths in metric units (meters and centimeters).

Default: NONMETRIC.

**THRESHOLD=n** Applies to: WRITE, READ

Specifies the maximum number of permanent and temporary errors which will be printed in the detail report for any given tape before printing is suspended. The summary report will still recap the total number of errors. This is used to limit the size of the printout.

Default: 1000 errors.

### 14.3 CONTINUED . . .

#### 14.3.5 CONTROL OPERANDS

**MODIFY=YES** Applies to: DEFAULT  
**NO**

Specifies whether FATS will respond to an operator's MODIFY (F) and STOP (P) console commands. FATS ordinarily issues a WTOR to the console at startup which the operator can reply to at any time to control FATS execution (see the WTOR parameter). If MODIFY=YES is specified the operator may use the OS MODIFY (F) and STOP (P) commands to control FATS processing.

Default: MODIFY=NO

**WTOR=YES** Applies to: DEFAULT  
**NO**

Specifies whether FATS will issue its WTOR message (FATSW02) in order to provide operator control over FATS processing. If MODIFY=YES is specified to allow operator control with OS commands, it is not necessary to have the outstanding reply available to control processing.

Default: WTOR=YES

See Section 13.1 for more information on the use of these operands.

#### 14.3.6 MISCELLANEOUS OPERANDS

**BLKSIZE=n** Applies to: WRITE

Specifies the size of the block which will be written to tape during certification, in bytes. "n" must be less than 65536 and more than twice the density (BPI) of the tape being certified (except for tape cartridges).

Default: 65535

**FILES=n** "n" specifies how many physical files FATS is to read from its input tape. A physical file is defined as all records up to and including one tape mark. If the input tape contains labels, the labels before and after each data file will each count as one file, so each labeled file is actually three files. If NOREWIND is specified, the header label file of the first data file will not be read. If "n" is 0, FATS will terminate when it reads two consecutive tape marks (with no intervening data); this is usually sufficient to read all data files on the tape.

Default: 3 files (one labeled file).

**WARNING:** If "n" specifies more files than are present on the tape, or if "n" is 0 and the tape does not end in two tape marks, the tape may physically run off the end of the reel and require operator intervention to rethread it (tape cartridges do not have this problem, but will get an error instead). Multi-volume datasets do not have double tape marks at the end except for the last volume. FATAR, the companion product to FATS, can properly read multi-volume tapes and is a better choice for most verification operations.

**NUMWRITES=n** Applies to: WRITE

Specifies the number of WRITE CCWs which will be chained together during a certify operation. In other words, this is the number of test blocks that will be written in one I/O. Increasing the value may improve certification performance but may impact other jobs using tape drives on the same string. The value can be from 1 to 50. WRITES to round (3420-type) tapes are never chained.

Default: 20

**14.3 CONTINUED . . .**

**RETCODE** Applies to: ALL

Causes FATS to terminate with a return code of 12 rather than abending in all cases where it would terminate with a U0888 abend code (due to major errors).

Default: Abend with U0888 if major errors occur.

**REWIND** Applies to: ALL

**NOREWIND**

REWIND (the default) causes FATS to rewind all tapes to load point before starting the specified operation, regardless of any LABEL= parameter in JCL. NOREWIND will tell FATS to leave the tape at the user specified position; for WRITE and ERASE statements, this can be used to certify/erase the remainder of a tape beyond existing files. NOREWIND resets all label options.

Default: REWIND

**WTO** Applies to: ALL

Causes FATS to report the results of each operation on the system console via a one line message indicating the total number of permanent and temporary errors, or abnormal completion of the operation.

**MODE=xx** Applies to: WRITE, LABEL, ERASE

MODE= is used on 9-track (round) tapes when the drive is capable of writing multiple densities, to choose the density at which the output operation is to be performed (density selection is automatic on input tapes). Valid values are:

MODE=CB (800 BPI)

MODE=C3 (1600 BPI)

MODE=D3 (6250 BPI)

If the drive is not capable of the density indicated, an error will occur.

Default: the highest density supported by the tape drive

**14.3.7 ANALYZE OPERANDS**

All of the operands which are valid on an ANALYZE statement input to FATAR (as described in a later section of this manual), are valid on a DEFAULT statement or an ANALYZE(n) statement when FATAR is invoked as a subtask of FATS.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

**16.0 FATS EXAMPLES OF USAGE****INTRODUCTION**

Examples are provided here to guide you in the use of FATS. As many of the common uses of FATS as possible have been included, and most of the control statements and parameters are illustrated here. However, be aware that they are just examples and must be customized to your installation and situation before use. Areas of such customization will probably include: dataset names, unit names for tape, special parameters for tape management systems. The examples assume that no special STEPLIB DD statement is required to execute FATS; this may not be true in your installation.

Because of the many possible combinations of operands, control statements, tape label types, etc., it is not practical to create examples which cover all potential FATS usage. Many examples illustrate more than one aspect of FATS use, e.g., two or more operands together. This does not mean that the two must always go together unless the accompanying text says so.

Many examples use LABEL=(,BLP) (bypass label processing) on the tape DD statements. For input tapes, BLP allows you to mount a labeled tape without knowing its volume serial or dataset name; for output, it allows creation of unlabeled tapes. Sometimes this is simply a convenience, but for many FATS operations, such as labeling and certifying new tapes, it is essential. In some installations, system parameters prevent the usage of BLP by most users; in this case, the FATS control statement parameter BLP can be substituted. See Section 11.8 for security considerations for the use of BLP.

**INDEX TO  
EXAMPLES**

The first-time or infrequent user of FATS should review many of the examples below since they frequently build upon one another in illustrating the usage and effects of various parameters. For your convenience, however, here is a list of the examples provided to aid in quickly finding the one you need. It shows the title of the example and the operations and keywords it illustrates.

<b>#</b>	<b><u>EXAMPLE NAME</u></b>	<b><u>OPERATION</u></b>	<b><u>KEYWORDS</u></b>
1	Certify SL tapes	DEFAULT WRITE	MULT,MODIFY
2	Certify SL tapes on multiple drives	WRITE	VOL
3	Certify and label tapes	WRITE	VOL,VOLINCR,NOSAVE, MAXVOLN,MAXERR, MAXCERR,NEWPAGE
4	Certify scratch tapes on 3 drives	DEFAULT WRITE	MULT,MAXVOLN,MAXCLEAN, METRIC
5	LABEL tapes	DEFAULT LABEL	LINECNT,OWNERID,VOL, VOLINCR,MAXVOLN,BLP
6	Label and certify new tapes	WRITE	VOL,VOLINCR,MAXCLEAN, WTO,NOSAVE
7	Visually examine bad tape	WRITE	STOP,VOL
8	Verify a SL tape	ANALYZE	
9	Verify a multi-file SL tape	READ	VOL,FILES
10	Verify on several drives	DEFAULT READ	VOL,FILES
11	Erase on multiple drives	DEFAULT ERASE	VOLINCR,MAXVOLN,VOL

**ISPF DIALOG  
SUPPORT**

**Most of the FATS functions can be performed by using the FATS ISPF interface.  
See Section 93 for further details.**

## 16.1 EXAMPLES

**EXAMPLE 1:** This example certifies any number of standard labeled tapes. The UNIT= parameter controls which type of tape drive is allocated for certification (e.g., 3480, 3490, 3590-1). Since no volume serial is specified, the operating system will ask for a scratch tape. With LABEL=(,SL) specified in JCL, tapes with any volume serials can be mounted. A tape management system, if present, can protect against accidentally overwriting active data by confirming the scratch status of any tape mounted. The MULT parameter will cause FATS to continuously request new scratch tapes until the system terminates FATS (See Section 13.1). Since the SAVLAB parameter is the FATS default, the original volume serial of each tape will be preserved. The PARM=I on the EXEC statement causes FATS internal documentation to be printed on SYSPRINT.

```
//FATS          EXEC   PGM=FATS,PARM=I
//SYSPRINT      DD     SYSOUT=*
//SYSUDUMP      DD     SYSOUT=*
//TAPE1         DD     DSN=FATS,UNIT=TAPE,
//              LABEL=(,SL),DISP=(,KEEP)
//SYSIN         DD     *
                DEFAULT  MODIFY=YES
                WRITE(1)  MULT
```

**EXAMPLE 2:** Certify 3 standard labeled tapes on two tape drives, specifying the required volume serials in FATS control statements via the VOL= parameter. UNIT=(3490,,DEFER) on the TAPE DD statements will prevent the operating system from requesting a mount until FATS passes it the required volume serial. This example certifies 3490E cartridges, but any valid tape device type can be substituted. Each tape must be previously labeled with the proper volume serial since FATS is opening them as labeled. Since the SYSPRIN1 and SYSPRIN3 DD statements are present, separate detail reports will be printed for each tape drive. Since the SUMMPRT DD statement is present, a separate summary report for all tapes on all drives will be printed.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT      DD     SYSOUT=*
//SYSPRIN1      DD     SYSOUT=*
//SYSPRIN3      DD     SYSOUT=*
//SUMMPRT       DD     SYSOUT=*
//SYSUDUMP      DD     SYSOUT=*
//TAPE1         DD     DSN=FAT1,UNIT=(3490,,DEFER),DISP=(,KEEP)
//TAPE3         DD     DSN=FAT2,UNIT=(3490,,DEFER),DISP=(,KEEP)
//SYSIN         DD     *
                WRITE(1)  VOL=111111
                WRITE(3)  VOL=(222222,333333)
```

**EXAMPLE 3:** Twenty tapes, with volume serials "AX0001" to "AX0020" are to be labeled and certified. Their current internal volume labels are unknown so label processing must be bypassed (LABEL=(,BLP) in JCL). VOL=AX0001 provides the first volume serial to be requested, VOLINCR=1 specifies that the serial is to be incremented by 1 each time, and MAXVOLN=20 limits FATS to 20 such volumes. The presence of VOL= and NOSAVE causes the new tapes to be re-labeled with the AXnnnn volume serials. NEWPAGE causes the detail report for each tape to start on a new page (for easier separation), and MAXERR=5 and MAXCERR=3 will cause any given tape to terminate if it has more than 5 permanent errors or a 3 inch continuous error.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT      DD     SYSOUT=*
//SUMMPRT       DD     SYSOUT=*
//SYSUDUMP      DD     SYSOUT=*
//TAPE5         DD     DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//SYSIN         DD     *
                WRITE(5)  VOL=AX0001,VOLINCR=1,NEWPAGE,NOSAVE,
                MAXVOLN=20,MAXERR=5,MAXCERR=3
```

## 16.1 CONTINUED . . .

**EXAMPLE 4:  
CERTIFY  
SCRATCH  
TAPES ON 3  
DRIVES**

94 scratch tapes are to be certified on three Magstar (3590) tape drives. The system operator knows which tapes to mount, so FATS is to simply call for scratch tapes. The DEFAULT statement is used so that the other FATS parameters only need to be specified once. MULT will cause FATS to repeatedly call for new tapes on each drive (since no volume serial is given in the JCL, scratches will be requested). MAXVOLN=31 causes each drive to terminate when 31 tapes have been certified on it (for the third drive, 32). MAXCLEAN=10 instructs FATS to ask the operator (via a console message FATSW06) to clean each tape drive between every set of 10 tapes (i.e., 3 cleanings per drive in this example) so that dirty tape drives don't cause false errors. Lengths will be reported in metric meters and centimeters.

**Note:** The SAVLAB parameter, which is the default, causes FATS to preserve the volume serials on the tapes; if any tape does not contain a label, FATS will ask the operator for the correct label.

```
//FATS      EXEC   PGM=FATS
//SYSPRINT  DD     SYSOUT=*
//SYSPRIN1  DD     SYSOUT=*
//SYSPRIN2  DD     SYSOUT=*
//SYSPRIN3  DD     SYSOUT=*
//SUMMPRT   DD     SYSOUT=*
//SYSUDUMP  DD     SYSOUT=*
//TAPE1     DD     DSN=FATS,UNIT=(3590-1,,DEFER),
//           LABEL=(,BLP),DISP=(,KEEP)
//TAPE2     DD     DSN=FATS,UNIT=(3590-1,,DEFER),
//           LABEL=(,BLP),DISP=(,KEEP)
//TAPE3     DD     DSN=FATS,UNIT=(3590-1,,DEFER),
//           LABEL=(,BLP),DISP=(,KEEP)
//SYSIN     DD     *
            DEFAULT  MULT,MAXVOLN=31,MAXCLEAN=10,METRIC
            WRITE(1)
            WRITE(2)
            WRITE(3)  MAXVOLN=32
```

**EXAMPLE 5:  
LABEL TAPES**

Volumes "TST990", "TST994", and "TST998" are to be labeled with those volume serials. The VOL= parameter specifies the first of those serials. The VOLINCR=4 parameter causes that serial (TST990) to be incremented by 4. Although MAXVOLN=50 is specified, the incrementing stops when the numeric portion of that serial cannot be incremented without overflowing. LINECNT=80 requests 80 lines per page on all FATS reports.

The BLP operand was specified on the DEFAULT statement. This can be used in installations where the LABEL=(,BLP) operand is treated as LABEL=(,NL) due to installation options. The FATS BLP operand forces BLP (bypass label processing) internally at OPEN time), overriding the JCL specification. See Section 11.8 for security considerations for the use of BLP.

```
//FATS      EXEC   PGM=FATS
//SYSPRINT  DD     SYSOUT=*
//SYSPRIN1  DD     SYSOUT=*
//SUMMPRT   DD     SYSOUT=*
//SYSUDUMP  DD     SYSOUT=*
//TAPE1     DD     DSN=FATS,UNIT=(TAPE,,DEFER),
//           LABEL=(,BLP),DISP=(,KEEP)
//SYSIN     DD     *
            DEFAULT  LINECNT=80,OWNERID='TEST TAPE',BLP
            LABEL(1)  VOL=(TST990),
                     VOLINCR=4,MAXVOLN=50
```

## 16.1 CONTINUED . . .

**EXAMPLE 6:  
LABEL AND  
CERTIFY NEW  
TAPES**

The tape librarian has 250 brand new tapes to label and certify (certifying is recommended to clean loose oxide from the new tapes) starting at volume serial 500001. The VOL= and VOLINCR= parameters will cause this to happen. Since no MAXVOLN= parameter was specified, FATS will call for up to 682 volume serials (up to 500682). However, when all tapes are done, or if the certification process must be stopped to do other work, the librarian can stop FATS from the MVS console (see Section 13.1). The WTO parameter will cause a 1-line summary of the errors on each tape to be written to the console so that potential problem tapes can be immediately identified. Since these are new tapes, the MAXCLEAN=5 parameter requests tape drive cleaning after every five tapes to remove accumulated oxide.

NOSAVE prevents reading blank tape and allows labeling to take place.

**Note:** to simply label the tapes without certification, change WRITE(1) to LABEL(1).

```
//FATS          EXEC   PGM=FATS
//SYSPRINT      DD     SYSOUT=*
//SYSPRIN1      DD     SYSOUT=*
//SUMMPRT       DD     SYSOUT=*
//SYSUDUMP      DD     SYSOUT=*
//TAPE1         DD     DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//SYSIN         DD     *
                WRITE(1)  VOL=500001,VOLINCR=1,MAXCLEAN=5,WTO,NOSAVE
```

**EXAMPLE 7:  
VISUALLY  
EXAMINE BAD  
TAPE**

The tape librarian wants to visually examine the bad spots on a known bad tape. The STOP parameter will cause FATS to stop and issue a message to the console when permanent data checks are encountered, at which time the tape can be examined. The operator may reply to the message, telling FATS to terminate, continue (and stop again on the next error), or turn off "STOP MODE" and continue normally. Stop mode might be used to determine if tape errors are due to damage or simply due to removable contamination.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT      DD     SYSOUT=*
//SYSPRIN1      DD     SYSOUT=*
//SUMMPRT       DD     SYSOUT=*
//SYSUDUMP      DD     SYSOUT=*
//TAPE1         DD     DSN=FATS,UNIT=(TAPE,,DEFER),DISP=(,KEEP)
//SYSIN         DD     *
                WRITE(1)  VOL=111111,STOP
```

**EXAMPLE 8:  
VERIFY A SL  
FILE**

A standard labeled tape "222222" is to be verified for readability. The FATS companion product FATAR is to be used to do so. Since FATAR is opening it as labeled, the proper dataset name and volume serial must be given. Since the FILES= parameter is not given, FATAR will read to the end of the tape (2 tape marks or EOVL labels).

```
//FATS          EXEC   PGM=FATS
//SYSPRINT      DD     SYSOUT=*
//SYSPRIN7      DD     SYSOUT=*
//SUMMPRT       DD     SYSOUT=*
//TAPESUMM      DD     SYSOUT=*
//SYSUDUMP      DD     SYSOUT=*
//TAPE7         DD     DSN=DATA.SET.NAME,UNIT=TAPE,
//              VOL=SER=222222,DISP=OLD
//SYSIN         DD     *
                ANALYZE(7)
```



## 16.1 CONTINUED . . .

**EXAMPLE 9:** A standard labeled tape "333333" containing two files is to be certified for readability. The volume serial is specified via the FATS parameter VOL=, but a dummy serial must be given on the TAPE DD statement to satisfy operating system requirements; UNIT=(TAPE,,DEFER) keeps the system from requesting the mount until FATS sets the actual serial. LABEL=(,BLP) is specified since the true dataset name of the first file is unknown. FILES=6 causes two standard label files to be read (the header and trailer label files count as files to FATS).

```
//FATS      EXEC   PGM=FATS
//SYSPRINT   DD     SYSOUT=*
//SYSPRIN1   DD     SYSOUT=*
//SUMMPRT    DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPE1      DD     DSN=DATA.SET.NAME,UNIT=(TAPE,,DEFER),
//           VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP)
//SYSIN      DD     *
READ(1)      VOL=333333,FILES=6
```

**EXAMPLE 10:** Several tapes are to be verified for readability using two tape drives. The VOL= parameter specifies which tapes are to be mounted on each drive. FILES=0 tells FATS that each tape is to read until two consecutive tape marks are encountered (two tape marks normally indicates end of data on the tape).

**WARNING:** *Tapes which are part of a multi-volume set (except the last) DO NOT end in two tape marks and may run off the end of the reel if read with FILES=0. FATAR is recommended to verify such tapes.*

```
//FATS      EXEC   PGM=FATS
//SYSPRINT   DD     SYSOUT=*
//SYSPRIN1   DD     SYSOUT=*
//SYSPRIN2   DD     SYSOUT=*
//SUMMPRT    DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPE1      DD     DSN=ANY.NAME,UNIT=(TAPE,,DEFER),
//           LABEL=(,BLP),VOL=SER=DUMMY1,DISP=OLD
//TAPE2      DD     DSN=ANY.NAME,UNIT=(TAPE,,DEFER),
//           LABEL=(,BLP),VOL=SER=DUMMY2,DISP=OLD
//SYSIN      DD     *
DEFAULT     FILES=0
READ(1)     VOL=(111111,222222,333333)
READ(2)     VOL=(444444,555555,666666)
```

## 16.1 CONTINUED . . .

**EXAMPLE 11:  
ERASE ON  
MULTIPLE  
DRIVES**

A large number of tapes are to be "Data Security Erased" before they are sold. Since they have consecutive serial numbers, the FATS parameters VOL=, VOLINCR=1, and MAXVOLN=50 are used to erase 50 tapes on each of 3 tape drives. FATS will call for the tapes by serial as needed. Although LABEL=(,BLP) has been specified to suppress operating system label handling, FATS will re-label each tape as it erases it. Note that "Data Security Erase" operations do not tie up the tape control unit, so that each of the three erase operations can proceed at full tape speed, and other tape jobs will not be impacted. Innovation strongly recommends that you run ERASE on a tape before writing data to be sent to a third party to insure that all corporate data left from previous use of the tape is removed.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT      DD     SYSOUT=*
//SYSPRIN1      DD     SYSOUT=*
//SYSPRIN3      DD     SYSOUT=*
//SYSPRIN5      DD     SYSOUT=*
//SUMMPRT       DD     SYSOUT=*
//SYSUDUMP      DD     SYSOUT=*
//TAPE1         DD     DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//TAPE3         DD     DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//TAPE5         DD     DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//SYSIN         DD     *
                DEFAULT VOLINCR=1,MAXVOLN=50
                ERASE(1) VOL=100001
                ERASE(3) VOL=100051
                ERASE(5) VOL=100101
```

**20.0 FATAR FUNCTIONAL DESCRIPTION****WHAT IS  
FATAR?**

FATAR (Fast Analysis of Tape And Recovery) is a multi-purpose magnetic tape utility, which can be executed under MVS/ESA and OS/390. FATAR can read any magnetic tape which is usable by those operating systems, and can process multiple files and multiple volumes in one execution. It offers the ability to summarize or examine in detail the data read from the tape. FATAR's error recovery procedures are often capable of reading data which causes data check errors when read by normal access methods.

FATAR can also create a copy of the input data on another tape, optionally applying user-specified modifications as it copies. FATAR never modifies the input tape, so creating a copy is the only way to recover from errors on the original input. FATAR can also be used to move data to new media (such as from round tapes to cartridges, or 3480 cartridges to higher density media such as 3590 Magstar).

In many installations, much of the installation's data is stored on magnetic tape, which is subject to many hazards. This data will be useful only if it was written correctly when first recorded and is still readable when needed. Many hours of man and computer time can be wasted trying to recover from physical (hardware) or logical (data) errors on tape. FATAR can greatly reduce this recovery time, or even discover problems before they occur.

**WHO USES  
FATAR?**

FATAR is ideal for use by tape librarians and operations personnel, by applications programmers, and by systems programmers.

**WHAT IS FATAR  
USED FOR?**

FATAR has many applications, including:

- Investigating an "unknown" tape, to discover its label type, file count, DCB characteristics, etc.
- Mapping a tape, providing a compact summary of the characteristics of all files on the tape.
- Examining the data on a tape.
- Verifying that a tape file is properly formatted (every block is checked against its DCB information).
- Detecting and correcting invalid variable spanned records.
- Verifying that certain data fields contain valid data.
- Scanning a tape for a certain type of data.
- Scanning a tape for read data checks, which may cause application job failures.
- Verifying the readability of critical or archive tapes.
- Correcting invalid data or data checks by creating a copy of the input tape(s) with the bad data corrected or dropped.
- Creating a good copy of a tape which was not properly closed (such as during a system failure).
- Recovering data from a tape partially overwritten in error with a shorter file.
- Creating a backup copy of any tape (or multi-volume set of tapes), even if multiple files exist on the tape.
- Replacing tapes with small blocksizes with copies using a larger blocksize (reducing elapsed and CPU times of applications which read that data).
- Creating a copy of a tape with a different label type from the original (such as AL-to-SL or NL-to-SL).
- Creating copies of tapes at higher density or on a different type of tape drive.
- Copying FDR/ABR tapes (similar to the operation of the FDR utility FDRTCOPY).
- Reading and copying tapes with blocksizes up to 256K (as supported by OS/390 2.10 and z/OS).
- Making an "image copy" (an exact bit-for-bit copy) of a tape volume.

## 20.0 CONTINUED . . .

**FATAR CONTROLS** FATAR has a flexible set of command statements entered as 80-character records, of which only columns 1-71 are used by FATAR. For many operations, the commands required are very simple since the default operations of FATAR are designed to handle the most common user requirements. However, the FATAR statements allow you to greatly modify FATAR operation, including the ability to specify an operation down to a specific block and record on the tape if required.

**INPUT TAPE PROCESSING** Unless you instruct it otherwise, FATAR will read all files on the tape volume (or multi-volume tape set) specified on its TAPEIN DD statement. If DCB information is available (from JCL or from the tape labels), FATAR will deblock each tape block into individual records if the record format is fixed or variable, allowing you to examine or modify each logical record. If you prefer, or if DCB information is not available, FATAR will treat each tape block as one record.

Since most tapes in IBM installations are labeled, FATAR will normally expect input tapes to contain labels (either IBM standard or ISO/ANSI), and will intelligently print the labels, extract dataset name and DCB information from them and treat only real data files as data. This will be true even if you have specified LABEL=(,BLP) in JCL, so that you can mount tapes whose volume serial you don't know or don't care about and still print and process their labels. However, FATAR will recognize unlabeled tapes automatically and treat all files on them as data. If you have a need to process label files as data, you can instruct FATAR to do so.

If the input data is in ASCII, FATAR will translate it to EBCDIC before processing. By default, the ASCII-EBCDIC translation table used includes only the most common characters. If you need translation of other characters, such as the Euro, see the ASCII= operand for custom translation tables.

If hardware data checks occur reading a block from a round (9-track reel-to-reel) tape, FATAR will perform a variety of RETRY operations to attempt to successfully perform the read. Even if all recovery is unsuccessful, FATAR will still make the block as read available for manual correction. On cartridge tapes, hardware error recovery is invoked to recover from data errors.

**DATA BLOCK PROCESSING** Once a block has been read, FATAR offers facilities via FATAR commands to:

- Print the block (in EBCDIC, HEXADECEMAL, or DUMP format (HEX and EBCDIC together) ). All or part of the block may be printed.
- Change the length of the block (extra bytes may be added at the beginning or end of the block or both).
- Change data in the block (either at a specified location or by scanning for a string).
- Scan specified fields in the block for valid packed or zoned decimal numbers (to help eliminate data exceptions in application programs).
- Scan specified fields in the block for validity by comparison to a user specified table of valid or invalid values.
- Scan for specific strings of data, in specified locations or in any location.
- Drop data (prevent copy to output tape).

If logical records are being deblocked, all of the above (except block length changing) can be applied to each logical record.

**NOTE:** Changes to the input blocks never affect the input tape. Modified blocks are written only to the output tape, if present.

## 20.0 CONTINUED . . .

**OUTPUT TAPE PROCESSING** If the output tape DD statement TAPEOUT is present, FATAR will copy each file read from TAPEIN to the output tape. This will be an exact copy of the input, unless you request modification of one or more data blocks, or dropping of certain data blocks. You may also request that entire files be dropped from the copy.

By default, any block which had an unrecoverable data check when read from TAPEIN will not be written to TAPEOUT, but you may override this.

FATAR will issue a standard OPEN and CLOSE for each file on TAPEOUT.

If the output tape is in ASCII format, the data blocks will be translated from EBCDIC to ASCII before they are written. If the records are variable format, appropriate conversion to or from the ASCII variable format and the IBM format will be done. By default, the EBCDIC-ASCII translation table used includes only the most common characters. If you need translation of other characters, such as the Euro, see the ASCII= operand for custom translation tables. Note that if both the input and output tapes are in ASCII format, data will be copied without translation but any data displayed or tested by FATAR will be translated to EBCDIC.

**FATAR PERFORMANCE** FATAR must read every block on the input tape(s) and, when copying, write each block to the output, its performance is dependent on several things, including:

- the blocksize of the data file (smaller blocks will take longer)
- the type of tape drive
- the type of channel on which the drive is attached (parallel or ESCON)
- whether the data is IDRC-compressed or not

To give you some idea of what to expect, we ran a few tests at Innovation using tapes containing an FDR disk backup (which uses large blocksizes, up to 56K). We used IBM 3480s on a parallel channel, and IBM 3590s (Magstar) on an ESCON channel. The 3480 backup was on 5 volumes, so tape rewind and mount times are taken out of these sample times. Both the 3480 and 3590 backups were IDRC-compressed. Each 3480 volume contained 300-400MB of data (before compression) and the whole backup was 1600MB.

- ANALYZE 1 3480 VOLUME (300MB) – 1.75 minutes (105 seconds, 2.8MB/sec)
- ANALYZE 1 3590 VOLUME (1600MB) – 5.5 minutes (330 seconds, 4.8MB/sec)
- COPY 5 3480 VOLUMES TO 1 3590 VOLUME (1600MB) – 8.5 minutes (510 seconds, 3.1MB/sec)

**These are sample times from a specific environment; your results will vary.**

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

**21.0 FATAR TECHNICAL SUMMARY****21.1 GENERAL****THE FATAR PROGRAM**

FATAR (Fast Analysis of Tape And Recovery) requires about 256K of memory, but a larger region size is recommended to accommodate special requirements and future enhancements.

FATAR **must** be linkedited and executed as an authorized program.

FATAR may also be called as a sub-task of its companion product FATS. This mode of operation allows multiple FATAR sub-tasks to operate in the same job, and also allows use of FATS keywords for specifying multiple executions of FATAR using the same input tape drive, but with different input tapes. In this case the memory requirements of FATAR will be the values listed above times the number of concurrent FATAR sub-tasks (plus the memory requirements of FATS itself, detailed elsewhere in this manual).

**CONTROL STATEMENTS AND REPORT OUTPUT**

FATAR accepts its control statement input from DD statement SYSIN. If no SYSIN DD is present in the step, all defaults are assumed. The control statements are 80-byte records, of which only columns 1 to 71 may contain information.

Report output is directed to DD statement SYSPRINT, which may be a SYSOUT file or any tape or disk file valid for output. SYSPRINT always has a logical record length (LRECL) of 121 bytes and writes Fixed Block (RECFM=FB) records. If SYSPRINT is directed to SYSOUT or if the blocksize is unspecified, 121 is assumed. The user may specify a blocksize for tape or disk output, but if it is not a valid multiple of 121, 1210 will be assumed.

SYSPRINT will contain a report identifying the tape or tapes read, including the volume serials, tape and drive type, and density (if applicable). For each file read, the header and trailer labels are displayed (if it is a labeled tape), and the number of records read are shown for each data file. Total data blocks and bytes are displayed for the entire tape. Any blocks which had data checks on the input are identified and displayed. If the input tapes are compacted by IDRC hardware, compaction information is displayed. FATAR control statements can request additional printout, such as printing data blocks or formatting label fields.

If the optional TAPESUMM DD statement is included, it will contain a page for each tape processed, with a 2 line summary of each file read, including information from the tape labels and a summary of the data read. DCB characteristics of TAPESUMM are the same as for SYSPRINT, described above. For compacted input tapes, IDRC statistics are displayed for each file and for the entire tape.

Section 25 contains examples of SYSPRINT and TAPESUMM output.

If FATAR is run as a sub-task of FATS, these differences apply:

- Printout will be directed to SYSPRINx (where "x" is the same as the digit on the FATS control statement ANALYZE(x)).
- Most ANALYZE operands can be specified on the FATS ANALYZE(x) statement, but the user can override those options or provide MODIFICATION/SCAN statements by supplying them in a DD statement SYSINx ("x" same as above).

**21.2 INPUT TAPE PROCESSING****INPUT TAPE  
TYPES**

The input to FATAR may be any one of the following:

- A standard labeled (SL) tape containing one or more files (datasets).
- A multi-volume set of standard labeled tapes containing one or more files.
- A tape with ISO/ANSI V3 or V4 labels (AL) containing one or more files in ASCII.
- A multi-volume set of ISO/ANSI V3 or V4 labeled tapes containing one or more files in ASCII.
- An unlabeled tape containing one or more files, recorded in EBCDIC or ASCII.
- An unlabeled multi-volume set containing only one file (unlabeled multi-volume multi-file tape sets cannot be supported by FATAR since it cannot tell how many files are on each tape).

FATAR can read tapes on any IBM or IBM-compatible tape drive including:

- IBM 3420 and 3422 9-track round (reel-to-reel) drives and equivalent drives from other vendors. Since these drives are capable of reading and writing at several densities (800 BPI, 1600 BPI and 6250 BPI), FATAR can read a tape written at a given density only if the assigned drive is capable of that density.
- IBM 3480 and 3490 tape cartridge systems (18 track) and equivalent drives from other vendors such as StorageTek
- IBM 3490E tape cartridge systems (36 track) and equivalent drives from other vendors such as StorageTek. Note that 3490E drives can read tapes written on 3480/3490 drives (18-track format) but cannot write that format.
- IBM 3590 (Magstar) tape cartridge systems (128 track) in either native or 3490E emulation mode.
- IBM 3590E (Magstar) tape cartridge systems (256 track) including the extended length cartridge.
- StorageTek Timberline 9490, 9840, T9940 and Redwood SD-3 tape cartridge systems
- Input tape cartridges may be compressed by the IDRC option or uncompressed (IDRC compression is also called "compaction")
- Input and output tapes may be in a Virtual Tape System (such as the IBM 3494 VTS). You can use the FATAR copy function to perform "virtual ejection" by copying a VTS virtual tape to a real tape volume outside the VTS.



## 21.2 CONTINUED . . .

**INPUT TAPE  
LABEL  
HANDLING**

Labeled tapes are handled by FATAR independently of the LABEL= parameter in the JCL. In other words, the TAPEIN DD statement may say

```
LABEL=( ,SL)  or LABEL=( ,AL)  or LABEL=( ,BLP)
```

and FATAR will still be able to recognize and process the labels on the tape. Standard labels (SL) and ANSI V3 or V4 labels (AL) will be detected even when operating system label processing is bypassed. If ANSI labels are found, it is assumed that the associated data file is written in ASCII code, and an ASCII-to-EBCDIC translation will be done on the labels and associated data files.

Unlabeled tapes, mounted with

```
LABEL=( ,NL)  or LABEL=( ,BLP)
```

are automatically recognized by FATAR. When it reads the first input block, FATAR sees that it is not a label and treats the tape as unlabeled automatically.

If LABEL=(,BLP) is permitted in your installation, it is often useful with FATAR since it allows you to mount a tape whose volume serial or first dataset name are unknown, or to set up a standard FATAR procedure which does not need to be modified for every run. Even if BLP is not permitted in JCL, FATAR has an operand BLP which invokes BLP processing internally. See Section 21.6 for security considerations for the use of BLP.

You can force FATAR to process a labeled tape as unlabeled so that all files on the tape (including files containing labels) are handled as data by FATAR, even if LABEL=( , SL) or LABEL=( , AL) is specified in the JCL (in this case, the tape is positioned so that the first file processed is the header labels of the file specified in the JCL).

If the LABEL= parameter on the TAPEIN DD statement specifies a file other than the first, e.g.,

```
LABEL=5  or LABEL=( 4 ,BLP)
```

FATAR will start its processing with that file. If the input tape is a labeled tape but you have specified BLP in the JCL, the file number should point FATAR to the header labels of some file on the tape (header labels normally start at files 1, 4, 7, etc.); if you point to a data file instead of a label file, FATAR will treat the tape as unlabeled.

Unless you override normal FATAR operation with the LABELS= operand on the FATAR ANALYZE statement, FATAR will automatically recognize the label type of the first input tape and automatically do the right processing. SL and AL tapes will be automatically identified, the labels will be formatted, and information from the labels will be used to process the input tape (and for creation of files on the output tape, if present). Unlabeled tapes will be automatically recognized and processed.

When reading header labels, FATAR will extract the dataset name (only the last 17 characters of the name are recorded in the label), and DCB information (RECFM, LRECL, and BUFOF (ANSI only)). These values will be used for opening the output tape file (if any) and for deblocking the blocks of the following data file. If you have the tape management interface enabled (see Section 21.5), FATAR will query it to get the full 44-character dataset name for all files on the input tape, and will use that name for output files.

When reading trailer labels, FATAR will verify the block count against the actual count of blocks read in the preceding data file, and produce a warning message if they do not match.

If a multi-volume labeled tape set is being read by FATAR, it will perform the processing described above on the EOVS labels of at the end of each tape and the header labels at the beginning of the next tape, but will process the file that crossed tapes as one large file.

21.2 CONTINUED . . .

**INPUT FILE HANDLING** For labeled tapes, FATAR will process all of the files on all of the volumes specified on the TAPEIN DD statement unless you deliberately limit it to a certain number of files with the NUMFILES= or ENDAFTER= parameters. FATAR will continue to read and process files until either:

- Two consecutive tape marks are read, with no data or labels between, but only if the two consecutive tape marks are found where header (HDRx) labels would normally be expected.
- The number of data files specified by NUMFILES= has been read.
- The block specified by the ENDAFTER= parameter has been read.
- EOVL labels at the end of a tape are read (indicating that the dataset is continued on another tape), but no more tape volume serials were specified.

This means that, when in label mode, FATAR can read and copy an entire multi-file, multi-volume tape set automatically.

For unlabeled tapes, FATAR will process multiple files on the input tape, but is unable to switch to new volumes since there is no way to tell how many files are on the current volume, and therefore when to switch volumes. By default, FATAR will process all files on one unlabeled tape volume until two consecutive tape marks are read (as for labeled tapes). FATAR will also stop when it encounters EOVL labels (it checks for EOVL even though labels are not being processed) since EOVL labels are followed by only one tape mark; this prevents FATAR from running off the end of round tapes or getting errors on cartridges. If NUMFILES= was specified, FATAR will stop after reading that many physical files.

For a truly unlabeled tape which is part of a multi-volume set, there is no way for FATAR to detect the end of the recorded data on the tape. In order to avoid an I/O error after the last data (and possibly running off the tape for round tapes), you must tell FATAR how many files exist on the tape with NUMFILES=.

**LABELS ONLY MODE** FATAR supports a mode where only label information is printed from the input tape. LABELS=ONLY operates much like LABELS=YES, except that data files will be rapidly skipped over using a Forward Space File (FSF) CCW. No output tape is allowed. This allows you to quickly display information about all labeled files on a tape volume (or multi-volume set).

**OS LABEL MODE** FATAR has a special label mode, LABELS=OS, where all label processing and tape positioning is left to the operating system. Input tapes are handled the same as any utility program or user program, except that they must be on tape. In this mode, FATAR will handle exactly one input file (NUMFILES=1 is forced), but that file can be multi-volume, either labeled or unlabeled, and can even be several files concatenated together, as long as all the files are tape. LABELS=OS can be used to bypass the restriction on multi-volume unlabeled tapes, or to create one output tape file from multiple input files.

**IMAGE COPY MODE** FATAR's special image copy mode, LABELS=IMAGE, is used to make an exact bit-for-bit copy of a tape volume onto another tape volume. Exactly one input volume (labeled or unlabeled) will be completely copied to the output tape, including copying all labels and data exactly as read from the input. However, image copy includes special functions, such as printing input tape labels (even though they are being copied as data) and copying empty data files (which would normally cause FATAR to stop prematurely).

## 21.2 CONTINUED . . .

**FATAR INPUT  
TAPE I/O**

FATAR uses EXCP rather than any standard access method when reading input tape (from DD name TAPEIN) in order to have more control over error recovery processes and to bypass some access methods restrictions. Prior to OS/390 2.10, normal access methods can only handled block sizes up to 32K; FATAR can handle blocks up to 64K, including those created by FDR and IBM's DFSMSdss). In OS/390 2.10 and z/OS, access methods support blocks up to 256K; these are now supported by FATAR.

FATAR will issue a normal OPEN for the first file on each input volume, allowing security and tape management systems to perform their normal checking and authorization. However, if a tape volume contains more than one file, FATAR will switch files without any intervening CLOSE/OPEN to improve performance. If a dataset occupies more than one input tape volume, FATAR will CLOSE at the end of the current volume and re-OPEN the next input volume using the current dataset name. This allows FATAR to process any multi-file and/or multi-volume tape or tapes in one execution. See Section 21.6 for security considerations on multi-file tape volumes.

If FATAR is invoked as a sub-task of FATS, it will read the input tape from DD name TAPEX rather than TAPEIN, where "x" is the same as the digit on the FATS control statement ANALYZE(x). This is compatible with the FATS DD name.

**EXTENDED I/O  
ERROR  
RECOVERY**

**NOTE:** Because of the extensive error recovery built into the hardware of cartridge drives (often using recovery methods not available to FATAR), extended I/O error recovery is not used on cartridge tapes and the following description does not apply to them. Data checks on cartridge tapes are considered to be permanent.

I/O errors on a tape are often fatal to an application program. If the operating system or tape hardware cannot recover from an error, the program has little choice except to abort and leave it up to a human to investigate and correct the problem. FATAR is an ideal tool for fixing these problems due to its extended error recovery and its response to unrecoverable errors.

FATAR's extended I/O error recovery is used on round (3420-type) tapes, and attempts to RETRY most errors in much the same way that the operating system's error recovery does. However, FATAR has the capability of retrying a read operation up to 256 times in both the forward and backward directions. Operating system error recovery does less than 1/6 of this and will not always RETRY in the backwards direction (particularly for large blocks).

To further improve recovery, FATAR will execute a special tape cleaner action on every fourth retry, moving the tape rapidly back and forth across the cleaner blade built-in to the tape drive, in order to attempt to dislodge any dirt or loose oxide on the tape surface.

Sometimes, the nature of an error on a particular data block is such that any attempt at recovery causes the tape to be mis-positioned, resulting in either lost data or an error recovery loop. Operating system error recovery cannot be totally shut off, and therefore cannot recover from this situation. FATAR can be instructed to do absolutely no recovery actions during a given run, and can often successfully bypass the bad block and recover the remainder of the data on the tape.

If FATAR's error recovery actions are unsuccessful, the data block is considered to be a permanent error. By default, such blocks will be dropped (not written) if you are copying to an output tape. However, FATAR gives the option of examining what data it was able to read and of modifying or replacing the invalid data. Most of the FATAR command statements can be specified to only apply to blocks which had a permanent data check.

Since FATAR must inhibit normal operating system error recovery in order to control its own error recovery actions, errors encountered by FATAR may not be recorded in the operating system "SYS1.LOGREC" (EREP File) or in SMF Type 21 (ESV – Error Statistics by Volume) records. For cartridge drives, the total errors encountered by the tape drive will still be recorded on LOGREC and SMF, but individual errors will not.

## 21.2 CONTINUED . . .

**INPUT TAPE  
DATA BLOCK  
PROCESSING**

As FATAR begins processing each data file, it analyzes the DCB information (record format and logical record length) it has available. If labels are being processed, FATAR will extract this DCB information from them. However, if DCB parameters were supplied on the DD statement for the input tape, they will override the tape labels. This allows you to suppress FATAR deblocking operations (by specifying DCB=RECFM=U) or to correct invalid DCB information on the tape labels.

If the record format of the file is fixed (RECFM=F or FB) or variable (RECFM=V, VB, VS or VBS), FATAR will attempt to deblock each data block it reads from the input tape into its logical records. Note that FATAR does not depend on the blocksize from the labels or JCL but will read all the data in each tape block and then attempt to deblock it according to the format and logical record length found. In fact, if the records are variable length, the record length from the labels/JCL is also not used, but rather the length fields at the start of each record are used to locate the records. If the record format is variable spanned (VS or VBS), FATAR will deblock each record segment individually.

If FATAR finds an error during deblocking (which may be a fixed length block which does not contain an exact number of full records, or a variable length block whose last record is too long for the length remaining in the block), FATAR will print an error message and treat that block as unblocked for printing. This may aid in identifying improperly formatted tapes, but remember that blocks that were read with data checks may cause these errors simply because the data is invalid.

When deblocking is in effect, FATAR will not print or make available for modification the block prefix (for ASCII Tapes) or the block descriptor word ("BDW" – for variable format blocks). However, each record (including the record descriptor word or "RDW" for variable format) is printed and available for modification.

The FATAR control statements SCAN and REPLACE (which validate and modify data respectively), will operate on individual records within the block, and can refer to specific records in specific blocks. The DROP statement can delete whole blocks or records within blocks. The PRINT statement always selects whole blocks but will print individual records within those blocks. The KEEP statement always operates on whole blocks.

If deblocking is suppressed or not possible, then the entire contents of each block (including any prefix or BDW) is printed or available for modification as if it were one single record.

**ASCII  
TRANSLATION**

If the input tape has ANSI labels, or if DCB=OPTCD=Q was specified on the TAPEIN DD statement, then all data records are translated from ASCII to EBCDIC as soon as they are read. By default, the ASCII-EBCDIC translation table used includes only the most common characters. If you need translation of other characters, such as the Euro, see the ASCII= operand for custom translation tables.

Conversely, if the output tape has ANSI labels, or if DCB=OPTCD=Q was specified on the TAPEOUT DD statement, then all records are translated from EBCDIC to ASCII before they are written. By default, the EBCDIC-ASCII translation table used includes only the most common characters. If you need translation of other characters, such as the Euro, see the ASCII= operand for custom translation tables.

However, if both the input and output tapes are in ASCII, the data will be copied without translation, so that no characters will be lost due to the translation tables.

The Euro and other special characters are not included in the default ASCII translation tables because the hex representation of the Euro and other characters will vary depending on the national character sets in use. To support these characters, you must build custom translate tables which will properly translate those characters in your environment. See the description of the ASCII= operand for details on custom translate tables.

## 21.3 MODIFICATION AND SCANNING

**MODIFICATION  
AND SCAN  
STATEMENTS**

FATAR MODIFICATION/SCAN Control Statements are edited by FATAR and stored in a table at initialization. As each block is read by FATAR, this table is scanned to see if any of these statements apply to the current block.

MODIFICATION/SCAN statements allow you to:

- Print the contents of any block
- Drop a block or record (suppress it from being written to the output tape)
- Keep a block (override FATAR's automatic drop of blocks which get data checks)
- Change the length of a block, making it shorter or longer than the original, and reposition the original data in the new block
- Replace data at a given position in a block or record, optionally verifying the original contents first
- Scan a block or record for any field containing a given value, and replace that with a new value (for either the first or every occurrence in the block or record)
- Validate that certain fields in a block or record contain valid packed or zoned (printable) decimal data
- Validate that certain fields in a block or record contain valid data bytes according to a user-specified test table (which can specify either all of the valid byte values or all invalid byte values)

More than one modification may apply to a given block or record, so FATAR will scan the table for all applicable control statements, executing them in the order that they originally appeared in the control statement input. Also, one statement may apply to more than one block or record; a range of blocks, or a range of records within a block may be specified, or a statement may apply to all blocks and records in a file (to provide a global MODIFICATION/SCAN facility).

Normally, all blocks which are modified will be printed after modification, although this can be suppressed. If fields are being validated (the SCAN statement), FATAR will PRINT a warning about each field which fails validation. In any case, a message will be printed about every block which was affected by a control statement (documenting that it was "DROPPED", "PRINTED", etc.). If the control input contains MODIFICATION/SCAN statements which are never selected by FATAR (if, for example, they specify a block number which does not appear on the tape), they will be identified at the end of the FATAR run so that you can investigate the reason.

Although the DROP, KEEP and REPLACE statements are most meaningful when an output tape is being produced, they will be processed normally even when no output tape is present. This allows you to "dry run" your control statements before you actually create an output.

**RENAME  
STATEMENT**

RENAME statements allow you to change the names of output files created by FATAR. They are processed like other MODIFICATION/SCAN statements and stored in the same table. However, they are used only at the point that an output tape file is being opened. Normally, FATAR will use the dataset name of the input file when opening the equivalent output file, but RENAME allows you to completely specify a new name (NEWN=), or specify a special string (NEWI=) which selectively modifies the input name, index level by level. Each RENAME statement can apply to one specific input file, or to all input files.

RENAME with NEWI= normally works only if the full dataset name of each input dataset is known (e.g., from tape management information), but the UNCOND operand on the RENAME statement will allow NEWI= to be used even when the full name is not known; the NEWI= will be applied to the truncated 17-character name from the tape labels (if the label name starts with a period, the period is discarded).

21.4 CONTINUED . . .

21.4 OUTPUT TAPE PROCESSING

**OUTPUT TAPE  
PROCESSING**

If the DD statement TAPEOUT is present, a logical duplicate of the input tape(s) will be generated. It is a "Logical" duplicate since FATAR will allow the copy to have a different type of labels from the original, tape density or tape type may be different, and user-specified modifications may be applied. The output tape will normally have the same number of files as were found on the input tapes, in the same order, with the same dataset names (the RENAME statement and the output exit allow you to rename files on the output tape).

As each file is encountered on the input tape, the equivalent file will be opened on the output tape; if the input tape contains 5 files, 5 files will be created on the output, with the same contents as the input. However, it is possible to request FATAR to drop an entire file (e.g., DROP LF=2,B=ALL) in which case the output file corresponding to the dropped input file will not be opened at all.

**OUTPUT LABEL  
PROCESSING**

Label creation on TAPEOUT is totally controlled by the LABEL= operand on the TAPEOUT DD statement. The tape will be created as standard label (SL), ANSI label (AL) or unlabeled (NL or BLP).

FATAR will generate a dataset name for each file opened on TAPEOUT; it may be recorded by your tape management system. If the output tape is labeled (LABEL=SL or AL), this name will also be recorded in the tape labels created. The name is derived by these rules:

- if the input tape is not labeled (or label processing is suppressed by LABELS=NO), the DSN= specified on the TAPEIN DD statement is used. However, if you have the tape management interface enabled (see Section 21.5) and the tape is recorded as unlabeled in the tape management data base, the dataset names recorded by tape management will be used.
- if the input tape is labeled and labels are read by FATAR, the dataset name is extracted from the HDR1 label that precedes each data file. However, although tape dataset names can be 44 characters, only the last 17 characters are recorded in the tape label. So FATAR does some additional processing to attempt to derive a proper dataset name:
  - if the name read from the label matches the last 17 characters of the DSN= parameter provided on the TAPEIN DD statement, that name is used.
  - if you have the tape management interface enabled (see Section 21.5), the input tape is recorded as labeled in the tape management data base, and the name from the HDR1 label matches the end of the full name of the equivalent input dataset from that data base, that full name is used.
  - if the EXIT= parameter was specified on the FATAR ANALYZE statement, a user-coded exit routine is called which can modify the output dataset name (described later in this section).
  - if a RENAME statement (Section 23.3.4) was specified which applies to this input file, the replacement name (NEWN=) or name modification string (NEWI=) will be used to name the output dataset.
- if none of the above provide the output name, the truncated 17 character name will be used (which may result in security errors).

If the output tape is labeled, the file sequence numbers of the output files created are controlled by the TAPEOUT DD statement. Normally, the first output file will be file sequence 1 regardless of the file sequence of the input file; the next output file will be 2, etc. If some input files are being dropped (via DROP LF=n,B=ALL) then no equivalent output file created; there will never be gaps in the file sequence numbers created on the output tape. If you want to use FATAR to add files to an existing tape, you can specify LABEL=n to specify the next available sequence number.

21.4 CONTINUED . . .

**OUTPUT LABEL  
PROCESSING  
(Continued)**

If labels are being processed on the input tape, DCB information read from the header labels of each file will be used to open the output file; if DCB= parameters are provided on the TAPEIN DD statement, they will override those from the tape labels. This DCB information may be recorded by your tape management system; if the output tape is labeled, DCB characteristics are recorded in the output tape labels.

**Note:** if DCB parameters are coded on the TAPEOUT DD statement, they will be ignored. The DCB characteristics of each output file are always derived from the current input file's labels or from DCB overrides on the TAPEIN DD statement.

When output files are closed, an option (CAT= operand) can be used to cause FATAR to catalog the each file it created on the output tape. This allows you to easily do a logical copy of any input tape and recatalog the datasets to the new volume. This is especially useful when copying tapes to a new media type (such as 3490 cartridge to 3590 Magstar cartridge).

However, if FATAR was unable to identify the full dataset name of the output dataset using the rules above, cataloging will not be done. Since the truncated dataset name from the tape labels is the last 17-characters of the name, it is not useful for cataloging. There are 2 exceptions:

- if NEWN= is used on a RENAME statement to provide a complete new name for a dataset being copied, CAT= will be honored.
- if the name from the input tape labels is 16-characters or less (blanks at the end), it is the complete name so CAT= will be honored.

**OUTPUT BLOCK  
PROCESSING**

FATAR uses EXCP to write blocks to the output tape, so the output blocks will be an exact duplicate of the input blocks in each file regardless of DCB characteristics, unless reblocking or modification is requested.

By default, any block which gets a permanent data check when read from the input tape will **not** be written to the output tape, under the assumption that the data that was read is probably not usable.

After all modifications have been applied to a given block from the input tape, it will be written to the output tape (unless it is being dropped). Standard system error recovery is permitted on the output tape, so any WRITE error reported back to FATAR will cause the entire run to be aborted at that point.

If the output tape has ANSI labels (LABEL=(,AL)), or if DCB=OPTCD=Q is specified on the TAPEOUT DD statement), the block will be translated from EBCDIC to ASCII; for variable format tapes, the block format is converted to ANSI variable format.

Dataset reblocking may be requested by the REBLOCK= keyword on the ANALYZE statement. If so, FATAR will reblock each copied file if it is fixed or variable format (except variable spanned and ANSI variable), if its logical record length is less than the new blocksize, and if its current blocksize is not greater than the value of MAXRBLK= (maximum blocksize to reblock) if specified. When reblocking, each logical record is copied into a reblocking buffer which is written to TAPEOUT only when full (or at end-of-file).

## 21.4 CONTINUED . . .

**FATAR OUTPUT  
FILE EXIT**

An optional user open exit may be used to modify the dataset names of the output files. If specified by the operand EXIT=(OPENEXIT,exitname), the module "exitname" will be loaded and will be called before opening each output file copied to TAPEOUT. If "exitname" cannot be found FATAR will abend with a S806 upon parsing the EXIT= keyword statement.

The register conventions when the exit is called are:

R1 – Pointer to PLIST containing:

- +0      Pointer to a fullword containing a function code:  
          4 is a normal entry, pointers at +8 to +16 are valid  
          8 is last call: allows you to clean up exit environment (such as freeing work areas).
- +4      Pointer to double word on double word boundary that can be used by the exit routine for any purpose such as remembering the subpool, length, and address of any area the exit acquired for operation.
- +8      Pointer to TAPEIN JFCB
- +12     Pointer to TAPEOUT JFCB
- +16     Pointer to a 44-byte field containing the dsname that FATAR intends to use for TAPEOUT.

R13 – Pointer to a standard save area

R14 – Return address

R15 – Entry point of exit routine

The registers should be saved and restored by the exit routine. Only the TAPEOUT JFCB should be modified; you can change the dataset name and/or dataset characteristics (RECFM, LRECL, etc.). The IBM macro IEFJFCBN can be used to define the JFCB fields.

Initially the dataset name in the TAPEOUT JFCB and the dsname field pointed to by the parmlist+16 are the same; both are provided to simplify exit coding, using the dsname field as input while you modify the JFCB dsname. If the tape management interface is in use (see Section 21.5), the dsname in the TAPEIN JFCB will be that determined by the interface. If the RENAME statement was used, the output dataset names will be the new name specified.

A sample exit routine (named "OPENEXIT") is included in the Installation Control Library (ICL) which was loaded from the FATS/FATAR installation tape (See Section 90). The sample exit replaces the high level qualifier (or everything in front of the first period) of the tape dataset name with a new qualifier hard coded in the exit. This shell could be expanded to add additional functions. However, note that the RENAME statement (see Section 23.3.4) is an easier way of renaming FATAR output datasets in most cases.

**MAKING AN  
IMAGE COPY**

FATAR has a special mode where it will create an "image copy" of a tape. Image copy will essentially create an exact bit-for-bit copy of an input tape on an output tape, including an exact copy of all of the labels on the input tape, so the output tape will even have the volume serial of the input (although an option is available to preserve the output serial).

This function is useful when you need to replace a tape that is damaged by copying all the data to a new volume; the new volume will be externally relabeled with the serial of the original tape and will take its place in the tape library. No catalogs or tape management information need be updated since the new tape looks just like the original.

Some years ago a tape vendor had to replace many thousands of tapes in hundreds of installations when it was found that manufacturing defects were causing the tapes to fail prematurely. FATAR image copy was used to copy many of the suspect tapes to new volumes transparently.



21.4 CONTINUED . . .

**MAKING AN  
IMAGE COPY  
(Continued)**

Image copy is invoked by specifying LABELS=IMAGE on the ANALYZE statement. LABELS=IMAGE is similar to the operation of the default LABELS=YES in that FATAR will process and display all labels on the input tape and will automatically switch to LABELS=NO mode if the tape is not labeled. However, LABELS=IMAGE causes some differences from normal processing in order to accomplish the image copy:

- the input tape will always be rewound and processed from the beginning, even if you have OPENed it to another position with the LABEL= parameter in JCL.
- only one input tape volume will be processed, even if you specify additional input volume serials.
- the OUTBLP option will be assumed, so that the output tape will be opened as an unlabeled tape. This is required so that FATAR can write the labels from the input tape onto the output.
- as labels and tape marks are read from the input tape, they will be written directly to the output (normally FATAR allows OPEN to create labels and tape marks on the output). FATAR will do only one OPEN on the output tape; it will not open every output file as it normally does.

The result is that the output tape will be a faithful, bit-for-bit image copy of the input tape volume (except that you can request that the output tape volume serial be preserved with the OUTSER=YES option). Of course, you can still use FATAR MODIFICATION statements to change data as it is being copied, but this is usually not done during an image copy.

Because the labels are copied as data, there are some restrictions when using image copy:

- only one tape will be read in a FATAR step doing image copy. If a multi-volume tape set must be copied, each tape must be copied in a separate step.
- for multi-volume tape sets, each tape volume must be completely copied to a separate output tape. Since all input volumes in a set (except the last) will contain data all the way to the end of the tape, you must be sure that the output tape used is long enough to contain all of the input tape data. Since tapes can come in various lengths, you must be sure that an output tape of the same length (or longer) is used. Even tapes that are supposed to be the same length may vary slightly, causing the copy to fail if the output is too short (some tape vendors provide "extra length" cartridges just to solve this restriction). To try to avoid this problem, FATAR will attempt to write output blocks beyond the normal logical EOT (End-of-Tape) during an image copy; for cartridges, it will continue writing until the drive reports it cannot continue (Physical EOT); if the difference in the tape lengths is slight, this is usually enough to allow the copy to complete.
- since FATAR does not OPEN each output file, they will not be recorded by your tape management system. For this reason, LABEL=EXPDT=98000 is usually coded on TAPEOUT (or whatever is required to bypass recording by your tape management system). Since image copy is usually used when a damaged or suspect tape is being copied to a new volume with the original serial number, this is not a problem since the new tape volume will simply replace the old volume in your tape library after being externally labeled with the input tape's volume serial; the existing tape management records are still valid for the new tape.
- no cataloging of the copied files can be done, but again, since the image copy will usually have the serial number of the input, the existing catalog records are valid and recataloging is not required.

**NOTE:**in previous releases of FATAR, image copy was invoked with LABELS=NO plus LABEL=(,BLP) on TAPEOUT. This can still be used, but the function provided by LABELS=IMAGE in FATAR V4.5 (and above) is easier to use, less error prone, and provides more information about the data being copied.

## 21.5 TAPE MANAGEMENT SYSTEM SUPPORT

FATAR V4.5 (and above) contains support to interface to tape management systems to get complete information about the names of datasets being read by FATAR. This is often necessary because the tape labels contain only a truncated version of that name (the **last** 17 characters of the original name).

Currently, CA-1 (TMS) and CA-DYNAM/TLMS, both from Computer Associates, are supported. As part of the installation of FATS and FATAR, you must assemble an interface module for your CA tape management system (see Section 90.3); it will need to be reassembled if you install a new release of tape management.

DFSMSrmm (also called just RMM) is IBM's tape management system. As of FATAR V4.7 it is also supported.

FATAR will determine which of these tape management systems you are executing and invoke the proper interface. If you have more than one tape management system in use, contact Innovation for assistance.

### USING THE TAPE MANAGEMENT INTERFACE

Invocation of the tape management support in FATAR is automatic. Once the true volume serial of the first input tape has been determined, FATAR will detect which tape management system you are using, if any, insure that the proper interface module has been assembled, and will interface to your tape management data base to query that volume serial.

The interfaces builds 2 tables used by FATAR:

- 1) a table of the full dataset names of each file on the tape. If the tape is part of a multi-volume tape set, all dataset names on the tape set are tabled.
- 2) if the tape is part of a multi-volume tape set, all volume serials which are part of the tape set. The label type (SL, AL, NL) and the tape status (active, scratch) of each tape is also in this table.

Of course, the interface may report that the input volume serial was not found in the tape management data base; this will be normal if the input volume is a foreign tape or if its true volume serial cannot be determined (for unlabeled tapes and tapes processed with LABELS=NO, FATAR will use the volume serial provided on the TAPEIN DD statement, which may not be the true serial). Other interface errors may prevent FATAR from getting the desired information.

### USING THE TAPE MANAGEMENT TABLES

As FATAR reads each file from the input tape, it will attempt to match that file against the table of dataset names, in order to get the full 44-character dataset name.

For labeled input tapes it will extract the truncated 17-character dataset name from the tape labels, and compare it against the end of the name recorded by tape management for this file. If they match, the full name is used by FATAR. If they do not match, the name from the labels is used. In either case, the name is reported in the FATAR listings.

For unlabeled input tapes, the name from the dataset table will always be used.

If the input is a multi-volume tape set, as each new tape is mounted FATAR will verify that its volume serial is the expected next volume according to the tape management volume table. If not, a warning message is issued, but processing continues. An option (NEXTVOL=TMS) is available which causes FATAR to mount and process the volumes from the volume table, ignoring the JCL; this allows you to specify only the first volume in a multi-volume tape set and still have FATAR process the entire set.

## 21.6 SECURITY

**PROTECTING  
FATAR  
EXECUTION  
AND OPTIONS**

FATAR allows you to use your security system to control which users are authorized to execute FATAR and which are allowed to use various FATAR options. For example, you might authorize all users to analyze input tapes, but restrict the ability to copy tapes. **These security checks (except for FATAR.DATASET.SECBYPAS) are disabled by default; see Section 90 for instructions on enabling them.**

Use of FATAR is controlled by resources defined in the RACF FACILITY class (or its equivalent in other security systems). Consult your IBM RACF or other security vendor documentation for the procedure for defining such FACILITY class resources. Once defined, you must grant READ authority to the resource for users authorized to use it.

For compatibility with earlier releases of FATAR, the default is to allow all users to execute FATAR and use all options. If your security system tells FATAR that a given resource name is not defined, all users will be able to use that function. If you define some but not all of the resources, only those functions are protected. For example, if you don't define FATAR.COPY, all users can use FATAR to copy tapes.

However, some security systems, such as ACF2, protect all resources by default. If these security checks are enabled in such systems, no one will be able to use these FATAR functions until they are defined.

The FACILITY class resources checked by FATAR are:

FATAR.ANALYZE	execute FATAR, users not authorized will be unable to use FATAR at all
FATAR.COPY	provide a TAPEOUT DD statement (copy a tape)
FATAR.LABELS.YES	use the LABELS=YES or REQUIRED option (the default)
FATAR.LABELS.NO	use the LABELS=NO option
FATAR.LABELS.IMAGE	use the LABELS=IMAGE option to image copy a tape
FATAR.LABELS.ONLY	use the LABELS=ONLY option to print only labels on a tape
FATAR.LABELS.EOD	use the LABELS=EOD option to recover overwritten cartridge tapes
FATAR.PRINT	use the PRINT statement to print data
FATAR.RENAME	use the RENAME statement to rename copied datasets
FATAR.BLP	use the BLP operand or LABEL=(,BLP) in TAPEIN JCL
FATAR.OUTBLP	use the OUTBLP operand or LABEL=(,BLP) in TAPEOUT JCL
FATAR.EOD	recover data past a hardware EOD mark on cartridge tapes
FATAR.DATASET.SECBYPAS	bypass dataset security checking for input files

As each control statement is processed, FATAR will issue a RACROUTE to verify that the user has at least READ authority to the proper resource name. The step will fail if RACROUTE returns a code of 8 (or higher) meaning that the resource is defined to the security system but the user associated with the FATAR step does not have authority to it.

## 21.6 CONTINUED . . .

**INPUT  
DATASET  
SECURITY  
CHECKING**

As FATAR opens each input tape, normal OPEN security checking is done, based on the parameters on your TAPEIN DD statement. However, if BLP is used, the dataset name you provide may not be the real dataset name on the tape; see notes below on the use of BLP.

For a multi-file input tape (TAPEIN), FATAR does one OPEN on the first file on each tape (or wherever the LABEL= operand on your TAPEIN DD points). For additional files, FATAR simply reads them, so no OPEN security check is done for those additional files. So, only the first file on each tape volume is security checked.

To close this security exposure, FATAR may issue a RACROUTE for the dataset name in the DATASET security class, to verify that the user has at least READ authority to the dataset name of each input dataset. This check is done only if FATAR has the full dataset name of the input dataset; this is usually true only if you have installed the FATAR Tape Management Interface (Section 21.5) and the input tape is recorded by your TMS. Note that in some cases (such as LABELS=NO and LABELS=IMAGE), the dataset names are not known and will not be checked.

Before issuing these checks, FATAR will check the FACILITY class resource FATAR.DATASET.SECBYPAS. If this resource is defined, and this user has at least READ authority to it, no DATASET checks will be done in this step. For compatibility with earlier releases of FATAR, if the resource is not defined to your security system, no DATASET checks will be done for any user.

**If you don't define resource FATAR.DATASET.SECBYPAS**, no additional dataset checks are done for any user; users may be able to access data with FATAR that they are not authorized to read. **If you do define it**, then dataset checks are done for *all users* except those whom you authorize to that resource.

**Note: The dataset security checks are done by default; see Section 90 for instructions on disabling then.** In security systems which protect all resources by default, such as ACF2, FATAR.DATASET.SECBYPAS will appear to be protected, so dataset security checks will be done for all users by default. If you don't want this, you must either disable the dataset security check or define the FATAR.DATASET.SECBYPAS resource.

**OUTPUT  
DATASET  
SECURITY  
CHECKING**

With the exception of image copy (LABELS=IMAGE), FATAR does a normal OPEN on every output dataset on TAPEOUT, which will cause normal OPEN security checking to be done. The user must be authorized to create a dataset with the name specified by FATAR. LABELS=IMAGE does an OPEN only on the first output file.

When processing with LABELS=YES, the dataset names extracted from the input tape will be used to open the equivalent output file. If you have the FATAR Tape management Interface installed (Section 21.5), this is normally the full 44-character dataset name.

If you do not have the TMS interface, the names may be the truncated dataset names from the input tape labels (*last 17 characters of the name*); since there will be no applicable security rule for this name, no security check will be done.

## 21.6 CONTINUED . . .

**USE OF BLP** When label processing on input or output volumes is bypassed using the BLP option in JCL or the BLP/OUTBLP operands on the ANALYZE statement, volume labels and dataset names are not verified by OPEN and so authorization to the volume and datasets cannot be guaranteed; if anything, your security system will check authority to the volser and dsname provided in the JCL when BLP is used, these may not match the actual tape. For this reason, most security systems allow your installation to restrict the use of BLP.

However, some functions of FATAR (such as mapping or copying a tape with an unknown volser or dsname and doing an image copy) **require** the use of BLP. Innovation suggests that you provide BLP authority to a limited number of users (operators, tape librarians, and/or system programmers) so that they can run these functions when required.

Resources FATAR.BLP and FATAR.OUTBLP control the ability to use the BLP (for TAPEIN) and OUTBLP (for TAPEOUT) operands on FATAR control statements and the ability to put LABEL=(,BLP) on those DD statements. When opening a tape, FATAR will check if BLP is being used, from either source, and will fail the step if the resource is defined but the user is not authorized. If the user is authorized to FATAR.BLP or FATAR.OUTBLP but you have a general control on the use of BLP defined in your security system, the user must also be authorized to that resource.

**READING DATA PAST EOD** FATAR.EOD controls the ability to read data past a hardware EOD (end-of-data) mark on a cartridge tape. The EOD mark was introduced with IBM 3490E drives and is used on all cartridge drives except 3480s. It is automatically written after the last file on the tape and normal access methods cannot read data past that mark; this is a security measure to prevent a user from writing a short file on a scratch tape and then accessing residual data past the EOD. In order to be able to recover data from accidentally overwritten tapes (see Section 24.2) FATAR can position past the EOD mark and read the residual data. If an EOD mark is encountered, FATAR will check for authority to FATAR.EOD before proceeding. You may want to protect FATAR.EOD to be sure that only authorized users can access data from overwritten tapes.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

**22.0 FATAR EXECUTION JCL**

To execute FATAR, the following JCL statements are required:

**ISPF PANEL  
SUPPORT**

**Extensive ISPF panel support is available to generate jobstreams and control statements for FATAR.**

**See Section 93 for further details.**

**EXEC  
STATEMENT**

The EXEC statement specifies the FATAR program name, memory requirements (if your installation defaults are insufficient), and parameter field.

A region of 0M is recommended for FATAR; the program will use no more memory than is required for a particular function.

The parameter field (PARM=) is used to define to FATAR the amount of memory required for storing FATAR modification and scan control statement information. FATAR requires about 30 bytes per control statement plus the data length of any D=, V=, or S= data fields given. The default table size of 32K bytes is sufficient for most FATAR applications, but if it proves to be inadequate, override it by PARM= 'SIZE=nnnnnn' where nnnnnn can be 1-999999.

Examples:

```
//FATAR EXEC PGM=FATAR,REGION=0M
//FATAR EXEC PGM=FATAR,PARM='SIZE=24576',REGION=0M
```

**STEPLIB/JOBLIB  
DD STATEMENT**

A STEPLIB or JOBLIB DD statement will be required if FATAR has been linked into a private library. It can be omitted if FATAR is in a system library which can be accessed without a STEPLIB/JOBLIB statement (that is, a library in the system link list). This **must** be an APF authorized library.

**SYSPRINT DD  
STATEMENT**

SYSPRINT receives all FATAR printout and reports (except the tape summary), and is normally allocated to a SYSOUT dataset. Its DCB attributes are RECFM=FBA,LRECL=121. If blocksize is specified it must be a multiple of 121, otherwise it will default to 121 for SYSOUT or 1210 for other devices.

**TAPESUMM DD  
STATEMENT**

TAPESUMM is optional, but if present receives a summary of all files on each tape processed, including dataset name, tape label information, block and byte counts, length of each file, minimum, maximum and average block sizes, and read error counts. Its DCB attributes are the same as for SYSPRINT.

**SYSUDUMP DD  
STATEMENT**

SYSUDUMP requests an abend dump if major errors occur (note that most internal abends in FATAR are for the user's information only and do not cause dumps). SYSUDUMP is usually allocated to SYSOUT. Abend dumps are necessary for analysis of problems by Innovation. If you have the ABEND-AID product from COMPUWARE include the following so that a fully-formatted dump is produced:

```
//ABNLIGNR DD DUMMY
```

## 22.0 CONTINUED . . .

**TAPEIN DD  
STATEMENT**

TAPEIN specifies the tape or tapes to be analyzed. If more than one volume serial is specified, the volumes should be part of a single multi-volume tape set, in the proper order. If the input dataset is cataloged, UNIT= and VOL= can be omitted, but be aware that for multi-volume, multi-file tapes, each catalog entry points only to the volumes that file occupies, so you may have to specify volume serials in the JCL in order to process the entire tape set. However, if you have the tape management interface enabled (see Section 21.5), you can code the NEXTVOL=TMS option which will use tape management information to mount and process all the volumes in the tape set automatically.

LABEL= can specify SL (the default), AL, NL, or BLP; FATAR's decision to process the tape as labeled or unlabeled is independent of JCL. However, the LABEL= parameter can be used to position the input tape to a particular file at which to start processing. If SL or AL is specified, the dataset name of the first file to be processed must be properly specified, and FATAR will position to the header labels of that input file before starting. Since this is an input file, DISP=OLD is appropriate.

If LABEL=(,BLP) is permitted in your installation, its use on TAPEIN is desirable to be able to analyze unknown tapes, or to analyze tapes without having to update the JCL with the proper volume serial and dataset name. Even if BLP cannot be specified in JCL, the BLP operand on the FATAR ANALYZE statement can be used to invoke BLP processing internally. See Section 21.6 for information on the security considerations for BLP.

If labels are being processed from TAPEIN, FATAR will usually receive sufficient information from those labels to deblock the data blocks in each file. However, if label information is missing or incorrect, it can be supplied by the DCB= parameter on TAPEIN. Even if label information is present, FATAR can be forced to treat input blocks as single large records for printing and modification by putting DCB=RECFM=U on the TAPEIN statement (the DCB information in the labels will still be used to open TAPEOUT files).

If LABEL=(,AL) or DCB=OPTCD=Q appears on the DD statement, or if FATAR detects ANSI labels on the tape, all labels and data read will be translated from ASCII to EBCDIC.

For image copy (LABELS=IMAGE), TAPEIN should specify only one volume serial; additional serials are ignored.

**EXAMPLES:**

```
//TAPEIN DD DSN=FILE.NAME,UNIT=TAPE,
// VOL=SER=111111,DISP=OLD
//TAPEIN DD DSN=ANY.FILE,UNIT=TAPE,
// VOL=SER=(22222,33333),
// LABEL=(4,BLP),DISP=OLD
//TAPEIN DD DSN=CATALOG.ED.DATASET,DISP=OLD
```



## 22.0 CONTINUED . . .

**TAPEOUT DD  
STATEMENT**

The TAPEOUT DD statement is optional. If TAPEOUT is omitted FATAR will simply analyze the input tape, but if present specifies a tape on which to create a copy of the input files read from TAPEIN. The LABEL= parameter on the TAPEOUT DD statement may be anything (SL, AL, NL, BLP) and FATAR will create the specified type of labels. Note that SUL or AUL is not required for the creation of user labels on TAPEOUT; if the input tape contains user labels, they will be copied to TAPEOUT (if labeled) unless the SUPULAB parameter is specified. Since this is an output file, DISP=NEW is usually appropriate. Volume serials may be specified, or may be omitted to call for scratch tapes.

If labels are not being processed on TAPEIN, the DSN= JCL parameter on the TAPEOUT DD is ignored and the DSN= specified on TAPEIN will be used to name every file created on TAPEOUT. If labels are processed on TAPEIN, the DSN= is still ignored, and every file name will be copied from TAPEIN labels. The RENAME statement can be used to override the output dataset names in both cases. If LABELS=OS is specified, the DSN= on the TAPEOUT DD **will** be used to name the output dataset.

Also, if LABELS=EOD is specified to recover data from an overwritten tape, the TAPEOUT DD statement must include a DSN= specifying the output dataset name for the first (partially overwritten) dataset to be recovered; additional recovered files will be named according to the names in their header labels or RENAME statements.

The LABEL=EXPDT= or LABEL=RETPD= parameter may be specified to provide an expiration date for the output files created. If omitted, the expiration dates of labeled files read from TAPEIN will be used.

If LABEL=(,AL) or DCB=OPTCD=Q appears on the DD statement, all data written will be translated from EBCDIC to ASCII.

The ability to provide a TAPEOUT DD statement (i.e., the ability to copy tapes with FATAR) may be restricted by security rules; see Section 21.6.

**NOTE:** if DCB parameters are coded on the TAPEOUT DD statement, they will be ignored. The DCB characteristics of each output file are always derived from the current input file's labels or from DCB overrides on the TAPEIN DD statement.

**EXAMPLES:**

```
//TAPEOUT DD DSN=NOT.USED,UNIT=TAPE,DISP=(,KEEP)
//TAPEOUT DD DSN=FILE.NAME,UNIT=CART,
//      VOL=SER=444444,DISP=(,KEEP)
```

**SYSIN DD  
STATEMENT**

The SYSIN DD statement is the source of FATAR control statements. It is normally a "DD \*" spool file, but can be any disk or tape file with DCB characteristics RECFM=FB and LRECL=80. It can also be omitted entirely, in which case FATAR will assume all defaults and no modification or scan statements.

## 22.0 CONTINUED . . .

**JCL WHEN** FATAR may be invoked as a sub-task of its companion product FATS in order to execute more than  
**FATAR IS** one FATAR task simultaneously in the same jobstep. In this case several differences from the  
**A FATS** above rules apply (in each case, the “x” referred to is the number “x” in the FATS control statement  
**SUB-TASK** ANALYZE(x)):

- FATAR will write its report to Print DD statement SYSPRINx rather than SYSPRINT (e.g., SYSPRIN1).
- The TAPEx DD statement rather than TAPEIN will be used for the input tape (e.g., TAPE1).
- If present, the TAPExOUT DD statement rather than TAPEOUT will be used for the output tape copy.
- Most ANALYZE statement parameters can be specified in the FATS ANALYZE(x) control statements. However, if you wish to override what FATS has specified, or wish to specify scan or modification control statements, use a SYSINx rather than a SYSIN DD statement (e.g., SYSIN1).

**23.0 FATAR CONTROL STATEMENTS****23.1 GENERAL****GENERAL  
RULES**

All FATAR control statements must be input on 80-character records of which only columns 1 to 71 are used by FATAR.

Every FATAR control statement must begin with an operation keyword from the list below. This keyword may begin in column 1 or be preceded by any number of spaces. It must be followed by one or more spaces. Every operation keyword accepts a number of operands which follow those blanks and are separated by commas with no intervening spaces. Control statements may be continued onto multiple input records by following the last operand with a comma and a space, and starting the next operand on the next record in any column (preceded by spaces).

Any FATAR control statement may contain comments; FATAR stops its scan of the control statement whenever it finds an operand followed by a space (or comma-space for continuation), so the rest of the statement may be filled with any comment desired. Any control statement record containing an asterisk ("\*") in column 1 will be bypassed by FATAR and may entirely contain comments.

FATAR operation keywords:

ANALYZE	Perform FATAR analyze/copy
COPY	Same as ANALYZE except that TAPEOUT is required
DROP	Do not copy one or more blocks
KEEP	Copy a block even if permanent data checks or change its length
PRINT	Print one or more blocks
REPLACE	Replace data in a block or record
SCAN	Validate data fields in a block or record for packed or zoned numeric or against a user-defined table
TABLE	Define valid characters for a scan operation
RENAME	Rename one or more files while copying

## 23.2 ANALYZE STATEMENT

ANALYZE COPY	ASCII=tabname	,MAXERR=nnn
	,BLP	,MAXTERR=n
	,BYTEFACTOR=1000 1024	,METRIC NONMETRIC
	,CAT=NO YES RECAT ONLY	,NEXTVOL=JCL TMS
	,CHAR HEX DUMP	,MSGLEVEL=0
	,CHARTAB=tabname	,NUMFILES NF=nnnnn
	,COMBFILES	,OUTBLP
	,COMPACTION=SYSTEM COPY COMP NOCOMP	,OUTDSN=YES
	,CRDATE=COPY TODAY	,OUTSER=YES
	,EMCCCUNIT=esoteric	,OUTVOL=INVOL
	,ENDAFTER=(LF=nnnnn,B=bbbbbb)	,PRINTTMS=YES NO
	,EODRETRY=nnn	,PRTLEN=nnnnnn
	,EOTBLOCKS=nnn	,REBLOCK=nnnnn,MAXRBLK=mmmmm
	,EOV=NO	,RETCODE
	,EXIT=(OPENEXIT,exitname)	,RETRY=nnn
	,EXPDAYS=nnn	,ROR=NO
	,KEEP	,STOP
	,LABELS=YES NO OS REQUIRED IMAGE ONLY EOD	,SUPULAB
	,LBLPRT=CHAR HEX DUMP FORMAT NONE	,VALIDATE=INPUT OUTPUT ALL INEXP
	,LENCHK	,VERIFY
	,LINECNT=nn	,WTO
	,MAJERR=nnn	

An ANALYZE or COPY Control Statement is required (unless FATAR control statement input is omitted altogether). However, all of its operands are optional. Operands may appear in any order on the statement. Operands are available to control the input tape operation, error recovery actions and printing plus a few others; they are presented in those categories below. One ANALYZE/COPY statement must be first in the input.

The COPY statement is the same as ANALYZE, except that it requires the presence of a TAPEOUT DD statement and always copies the input tape.

**23.2 CONTINUED . . .**

If ANALYZE/COPY has no operands, or if the control statement dataset is omitted altogether, FATAR will perform the following functions:

- Input tape will be tested for presence of labels. If not labeled, all files on one tape will be read up to a double tape mark (end of tape). If labeled, all files on all input volumes will be read until a double tape mark is read or the list of volumes is exhausted (EOV labels will cause tape switching).
- If labels are ANSI, all labels and data will be translated from ASCII to EBCDIC.
- If present, all labels will be printed. If the TAPESUMM DD statement is present, a summary of the files read will be printed for each input tape.
- All temporary and permanent data checks will be noted, including printing 80 bytes from the blocks preceding and following the data check block.
- If DCB information is available (from tape labels or JCL), all blocks on TAPEIN will be checked to ensure that they don't exceed the blocksize, that they contain validly formatted fixed or variable records, and, for variable spanned records, that the record segments are valid.
- If the TAPEOUT DD statement is present, all data read from TAPEIN will be written to TAPEOUT, except that data check blocks will be dropped. The label type of TAPEOUT is controlled by the JCL for TAPEOUT. If TAPEIN is labeled, the output datasets will have the same names as the input datasets.

**23.2.1 INPUT TAPE OPERANDS**

**ASCII=** Specifies the name of a load module containing a pair of 256-byte translate table, one for translation of ASCII data to EBCDIC, the other for translating EBCDIC to ASCII. This will be used by FATAR whenever ASCII translation of input or output data is required. It can be used to include special characters, such as the Euro, in the translation. The load module will be LOAded by FATAR so it must be in the FATAR load library or some other library accessible to FATAR. See Section 90.4 for instructions on creating your own translate table.

Default: an IBM translate table is invoked via the IBM XLATE macro. This IBM-provided table includes only common ASCII and EBCDIC characters and excludes most special characters (such as the Euro).

**BLP** The BLP operand is provided to help when installation conventions do not allow LABEL=(,BLP) to be specified on a JCL DD statement, but it is desired to bypass label processing on TAPEIN so that FATAR can open and process any tape (such as when the true volume serial is unknown). When BLP is specified on the ANALYZE/COPY statement, the JCL label parameter will be changed to BLP internally before TAPEIN is opened. If a file number is specified in JCL it is not changed, e.g., LABEL=3 will become LABEL=(3,BLP). See Section 21.6 for security considerations for the use of BLP.

**EMCCCUNIT=** Specifies the esoteric name (1-8 characters) assigned to virtual tape drives managed by the EMC CopyCross product (a software virtual tape system), for example, EMCCCUNIT=VT90. Tape datasets cataloged to the device type associated with this esoteric name will be identified as VIRTUAL in reports.

**ENDAFTER=** The ENDAFTER operand causes FATAR to stop reading TAPEIN after a certain block in a given file has been processed; it is much like the NUMFILES= parameter above except that it can terminate at any point within a file. "nnnnn" is the logical file number (see "MODIFICATION/SCAN OPERANDS" in Section 23.4) and "bbbbbb" is a decimal number giving the block number (the first block in each file is block 1).

"n" may be 1-32767, "b" may be 1-2147483647.

## 23.2 CONTINUED . . .

**EOV=NO** If LABELS=NO has been specified to suppress processing of TAPEIN tape labels but the tape is actually labeled (such as when doing an “image copy”), FATAR will still perform 2 label checks: it will identify EOV labels since they are **not** followed by two tape marks, and FATAR cannot otherwise tell when it has reached the end of the tape. Also, if copying to an output tape, block counts will be updated in any EOF or EOV labels encountered. Specifying EOV=NO suppresses both of these actions.

**LABELS=** Controls processing of tape labels on TAPEIN. Use of the various values for LABELS= may be restricted in your installation by security rules; see Section 21.6. The examples in Section 26.2 will help you understand when to use each option.

**YES** specifies that the input tape is to be checked for standard IBM labels or ISO/ANSI labels; if present, these labels will be processed and printed, and the dataset name and DCB information will be used for copying the dataset; if not present, the tape will be treated as unlabeled (as if LABELS=NO was given).

**NO** forces the input tape to be treated as unlabeled (even if it does have labels); all physical files will be treated as data. This can be used on a labeled tape to perform an “image copy”.

**OS** requests that the operating system is to handle all label processing instead of FATAR. In this mode FATAR does a analyze or copy of a single file only. NUMFILES=1 is assumed and JCL parameters control whether TAPEIN is treated as labeled or not.

**REQUIRED** is the same as LABELS=YES, but in addition, if VALIDATE=ALL or INPUT is specified, the input tape will be rejected if it is not labeled.

**IMAGE** is invokes a special “image copy” mode, used to make an exact bit-for-bit duplicate of a single input tape volume on an output tape. It acts similar to LABELS=YES for processing of labeled and unlabeled input tapes. See Section 21.4 for details.

**ONLY** is similar to LABELS=YES except that no output tape is allowed and data files are rapidly skipped over. It provides a way to quickly print information about all labeled files on the tape without having to analyze the associated data files.

**EOD** is a special copy mode for recovering data from overwritten tapes. All cartridge drives except 3480 write a EOD (End-of-Data) mark after the last block written to the tape; the residual data on an overwritten tape is beyond the EOD. LABELS=EOD causes FATAR to search for the first EOD mark on the tape and recover the data beyond it. If the first data it can read past the EOD is a header label, it will process in LABELS=YES mode. If the first data is a data record, it will process that partial file in LABELS=NO mode, but if there are additional labeled files on the tape past the overwritten file, it will process them in LABELS=YES mode. Note that FATAR may not be able to determine if the data beyond the EOD was compressed; if you want the output tape to be compressed you should specify COMPACTION=COMP.

Default: LABELS=YES

**NOTE:** LABELS=YES will be automatically changed to LABELS=NO if the first block read is not a label; this means that LABELS= rarely needs to be used since FATAR will properly handle a labeled or unlabeled tape automatically.

## 23.2 CONTINUED . . .

**NEXTVOL=** Controls how FATAR determines the next input volume for multi-volume input tape sets.

**JCL** causes FATAR to mount and read all of the volumes specified by the TAPEIN DD statement. These volumes must all be part of a multi-volume tape set, in the proper order. Note that if the tape set contains multiple datasets (multi-volume multi-file) and the TAPEIN DD references a dataset on the tape via the system catalog (UNIT and VOLUME not given), it may not mount all the tapes in the set; in this case you will usually have to specify all the volumes in the set in the JCL.

**TMS** causes FATAR to use the table of tape serials obtained from your tape management system to mount and read all the tapes in the multi-volume set. The first tape is determined by the TAPEIN DD statement, but any other volumes which follow that first volume are taken from the tape management information. If the FATS/FATAR tape management interface has not been installed (See Section 90.3) or if the first input tape is not recorded in your tape management data base, it will revert to NEXTVOL=JCL processing.

Default: NEXTVOL=JCL

**NUMFILES=** "n" is a decimal number specifying the number of logical files which FATAR is to read from TAPEIN. A logical file is one standard label file if labels are being processed, or one physical file if not. If labels are processed and an input file extends to more than one volume, it is still considered one logical file. If "n" is 0, FATAR will read all files until two consecutive tape marks are read (signifying the logical end-of-tape), or until all input tape volumes have been processed. Also, if labels are not being processed and "n" is 0, FATAR will stop if EOVS labels are read. Note that if "n" is a value greater than the actual number of files on a tape, round tapes may run off of the reel and require operator intervention to rethread; this cannot occur on cartridge tapes. "n" may be 1-32767.

Default: NUMFILES=0

**PRINTTMS=** If the FATS/FATAR tape management interface is installed, FATAR will query your tape management system using the volume serial of the first input tape; the interface will return a table showing all datasets on the tape (or tape set) and all volumes in the tape set (See Section 21.5).

**YES** will print the tables returned by the tape management interface.

**NO** will not print the tables.

Default: PRINTTMS=NO

**STOP** When specified, message FATSW12 will be written to the operator at the end of each file read from TAPEIN, giving the option of terminating FATAR or continuing. This provides a measure of dynamic control of FATAR.

**WTO** When specified, message FATSW11 will be written to the operator at the end of each file read from TAPEIN or verified from TAPEOUT, giving the block and error counts.

## 23.2 CONTINUED . . .

## 23.2.2 ERROR RECOVERY OPERANDS

**EODRETRY=** "n" is a decimal number specifying the number attempts that FATAR will make to reposition and continue after an EOD (End of Data) mark is encountered on cartridge tapes. 3490E, 3590 (Magstar) and similar drives record an EOD mark after the last valid data on an output cartridge; EOD marks are usually encountered by FATAR only when trying to recover data from an overwritten tape; the EOD mark precedes the data to be recovered. Note that 3480 drives do not record an EOD mark. Depending on the type and vendor of the drive, FATAR will make the specified number of attempts to position past the EOD and read the overwritten data; it may not be successful, and even if successful may be forced to space past some of the original data. "n" may be from 0 to 255.

Default: EODRETRY=0

**KEEP** By default, FATAR will not copy to TAPEOUT any blocks which had permanent data checks when read from TAPEIN. If KEEP is specified, FATAR will write to TAPEOUT whatever data it was able to read from TAPEIN during the last recovery attempt on a data check block. Note that the block length may be incorrect due to the data check.

**MAJERR=** If "n" major errors (FATS046/7/8/9 messages) are detected on TAPEIN, FATAR will issue message FATS016 and abend with a U0888 Abend. "n" may be 1-32767.

Default: MAJERR=1

**MAXERR=** If "n" permanent data checks are detected on TAPEIN, FATAR will abend with a U0888 Abend. "n" may be 1-32767.

Default: MAXERR=20

**MAXTERR=** If "n" temporary data checks are detected on TAPEIN, FATAR will abend with a U0888 Abend. A temporary data check is one which FATAR reads successfully during recovery processing. "n" may be 1-32767.

Default: MAXTERR=100

**RETRY=** "n" specifies the number of times FATAR will attempt to re-read a block on TAPEIN which results in a data check before considering the error permanent. FATAR will attempt this many retries in each direction (forward read and backward read). "n" must be from 0 to 256. "RETRY=0" suppresses all re-read attempts and considers the error permanent immediately; if a tape drive cannot seem to "get past" an error on a tape, this may allow the tape to be read. RETRY= is used only on round (3420-type) tapes; for tape cartridge systems, because of the extensive error recovery done by the tape control unit, RETRY=0 will be forced and cannot be overridden.

Default: RETRY=40 on round tapes, RETRY=0 on cartridges

**ROR=NO** ROR=NO suppresses "READ-OPPOSITE-RECOVERY", the attempt to recover from a data check by reading it in the backwards direction. With this parameter specified, FATAR will only attempt forward re-reads for data checks. ROR is used only on round (3420-type) tapes.



## 23.2 CONTINUED . . .

## 23.2.3 OUTPUT TAPE OPERANDS

**COMPACTION=** If TAPEOUT is a cartridge tape capable of writing compacted (IDRC) format, COMPACTION= (which can be abbreviated COMP=) controls compaction on the output tape.

**SYSTEM** uses the system default for your installation.

**COPY** will use compaction if the input file was compacted, otherwise the output file is not compacted.

**COMP** will always compact the output tape.

**NOCOMP** will never compact the output tape.

Following MVS standards, labels on cartridge tapes are never compacted by FATAR.

Default: COMPACTION=COPY. However, if LABELS=OS is specified, compaction is controlled by the TAPEOUT DD statement (if the TRTCH=COMP or NOCOMP operand is not specified on the DD statement, the system default will be used).

**Note:** if you are recovering data beyond end-of-data with LABELS=EOD, FATAR may not be able to determine if the recovered data was compressed, so you may need to specify COMPACTION=COMP if you want the output tape to be compressed.

**CAT=** Controls cataloging of tape files copied to the output tape:

**NO** – output tape files will not be cataloged.

**YES** – output tape files will be cataloged if they are not already cataloged.

**RECAT** – output tapes files will be cataloged; if they are already cataloged, the catalog will be updated.

**ONLY** – output tape files will be cataloged only if the equivalent input tape files are cataloged (the catalog must point to the matching volser and file sequence number of the input file).

If you are copying a tape and are not renaming the output files (no RENAME statement), CAT=RECAT or CAT=ONLY is required to change the catalog to point to the new output volume and device type.

Cataloging will be attempted only if FATAR knows the full output dataset name; in most cases this is true only if the FATS/FATAR tape management interface is enabled; see Sections 21.4 and 21.5 for details. Cataloging is ignored for an image copy (LABELS=IMAGE).

Default: CAT=NO

**CRDATE=** If TAPEOUT is a labeled tape, the creation date recorded in the labels on TAPEOUT will be the current run date, unless CRDATE=COPY is specified, causing the creation date to be copied from the corresponding labels on TAPEIN if it is also labeled.

CRDATE=COPY is not recommended for CA-1 (TMS) tape management system users since it affects the expiration date on TAPEOUT in unpredictable ways.

Default: CRDATE=TODAY

## 23.2 CONTINUED . . .

**EOTBLOCKS=** When performing an image copy (LABELS=IMAGE), the output tape volume must always be long enough to hold all of the data on the input tape volume. However, when the input volume was originally written all the way to the end (usually because it was part of a multi-volume tape set), minor differences in tape lengths may make the chosen output tape just slightly too short to hold all the input data before logical EOT (End of Tape) is reached. EOTBLOCKS= specifies the number of additional blocks FATAR will attempt to write to TAPEOUT even after reaching logical EOT. Care must be used on round (3420-type) tapes since writing too much data after EOT may cause the tape to run off the reel and require operator intervention; this cannot occur on cartridge drives.

Default: EOTBLOCKS=10 on round tapes, EOTBLOCKS=999 on cartridge tapes.

**OUTBLP** OUTBLP is provided to help when installation conventions do not allow LABEL=(,BLP) to be specified on JCL DD statement, but it is desired to bypass label processing on TAPEOUT so that FATAR can open and process any tape (such as when the true volume serial is to be overwritten in an image copy). When OUTBLP is specified on the ANALYZE/COPY statement and DEFER is coded in the UNIT parameter of the DD statement, the JCL LABEL parameter will be changed to BLP internally before the tape is opened. If a file number is specified in JCL it is not changed, e.g., LABEL=3 will become LABEL=(3,BLP). See Section 21.6 for security considerations for the use of BLP.

OUTBLP is forced if LABELS=IMAGE is specified, since BLP processing on the output tape is required for image copies (See Section 21.4).

**MAXRBLK=** "n" is a decimal number (4-262144) giving the largest blocksize to be considered for reblocking (MAXRBLK= is meaningful only if REBLOCK= is also specified). Any input file whose blocksize on TAPEIN is greater than "n" will not be reblocked.

Default: All files are reblocked if REBLOCK= is specified.

**OUTDSN=YES** When LABELS=IMAGE is specified (an "image copy"), and VALIDATE=OUTPUT or ALL is specified, OUTDSN=YES requests FATAR to verify that the dataset name in the labels of the output tape consists of 17 character zeros ("0"). This is the state that a tape freshly initialized by FATS or by IEHINITT will be in, so this verifies that the output tape is a newly initialized tape.

Default: the output dataset name is not verified.

**OUTSER=YES** When LABELS=IMAGE is specified (an "image copy"), OUTSER=YES will cause the volume serial of the output tape mounted to be preserved, i.e., the image copy is modified so that the output tape volume serial is inserted in the volume label (VOL1) and dataset labels as the data is copied. In most cases, this type of copy should be done as a logical copy (LABELS=YES), but OUTSER=YES can be used when the exact layout of the tape must be preserved.

Default: the volume serial of the input tape will be copied to the output during an image copy.

**OUTVOL=INVOL** When LABELS=IMAGE is specified (an "image copy"), and VALIDATE=OUTPUT or ALL is specified, OUTVOL=INVOL requests FATAR to verify that the tape mounted on TAPEOUT has the same volume serial as the tape on TAPEIN, in other words, that the output tape has been initialized and labeled with the same volser as the input tape in preparation for the image copy.

Default: the output tape serial will be used for validation (if requested).

## 23.2 CONTINUED . . .

**REBLOCK=** "n" is a decimal number (4-262144) specifying the target blocksize for data reblocking. When this keyword is specified, all files will be reblocked as they are copied to have blocksizes as close to "n" as possible. MAXRBLK= may be specified to bypass reblocking on files which already have an acceptably large blocksize.

**Note:** REBLOCK= values over 32760 should be used only on systems which support blocksizes over 32K, such as OS/390 2.10 and z/OS, which support blocks up to 256K (262144). Whenever you reblock tapes, you must be sure that the applications which will read the tape are capable of handling the new blocksize.

Default: Blocks are copied exactly as read from TAPEIN, without reblocking.

**SUPULAB** If any file read from the input tape contains user labels and the output tape is a labeled tape, the user labels will be copied to the output tape unless SUPULAB is specified.

Default: User labels copied if output tape labeled.

## 23.2 CONTINUED . . .

## 23.2.4 PRINT CONTROL OPERANDS

**BYTEFACTOR=** Whenever byte counts are rounded to kilobytes or megabytes, BYTEFACTOR specifies the divisor used to calculate the values. Only 2 values are accepted:  
 BYTEFACTOR=1000 (the default) divides the byte counts by 1000 or 1000000.  
 BYTEFACTOR=1024 divides by 1024 or 1048576.

Default: BYTEFACTOR=1000

**CHAR** **CHAR** specifies that FATAR will print any required blocks or records in EBCDIC, 80 characters per print line.

**HEX** **HEX** specifies that FATAR will print in hexadecimal format, 8 groups of 4 bytes (32 bytes total) per print line.

**DUMP** **DUMP** specifies a combination, with the 32 characters of EBCDIC printed to the right of the same 32 bytes of data in hexadecimal.

This format can be overridden for specific blocks on a PRINT Control Statement.

Default: CHAR

**CHARTAB=** Specifies the name of a load module containing a 256-byte translate table. This will be used by FATAR whenever it translates hexadecimal data to EBCDIC for printing. This can be used for special printer character sets or for special national characters. The load module will be LOAded by FATAR so it must be in the FATAR load library or some other library accessible to FATAR. See Section 90.4 for instructions on creating your own translate table. The translate table name can be anything you like, except that FATCHRTB cannot be used.

Default: an internal translate table (FATCHRTB) is used which prints standard US printer numerics, uppercase alphabets and normal special characters. All other characters are translated to periods.

**LBLPRT=** Whenever FATAR recognizes labels on the input tape, it will normally print those labels (even if label processing has been suppressed by LABELS=NO). Formatting of labels is independent of the CHAR, HEX, and DUMP operands described above. LBLPRT= specifies an overriding print format for labels only. The values CHAR, HEX, and DUMP have the same meanings as the equivalent operands described above.

**LBLPRT=FORMAT** causes the labels to print in character format, followed by a two line breakdown of the individual fields in each label so that the contents of those fields can be determined more easily. **LBLPRT=NONE** suppresses label printing. Labels will be recognized and printed even when label processing has been suppressed by LABELS=NO.

Default: LBLPRT=FORMAT

**LINECNT=** "n" specifies the number of lines per page (1-32767) to print on all FATAR print datasets.

Default: LINECNT=60

## 23.2 CONTINUED . . .

- MSGLEVEL=0** By default, FATAR will print the contents of any block which has been modified by a REPLACE or KEEP Statement (up to the limit specified by PRTLEN=). MSGLEVEL=0 will suppress this print (FATAR will continue to print a line documenting that the block was modified).
- PRTLEN=** Whenever a block from the input tape is to be printed, for whatever reason, FATAR will print only the first "n" bytes or characters of data (0-262144) unless overridden by a PRINT Control Statement. If the block is being deblocked, FATAR will print as many logical records as exist in the first "n" bytes (and may truncate the last record).  
Default: PRTLEN=80
- NONMETRIC** When NONMETRIC is specified (or defaulted) FATAR will list tape lengths and error positions and lengths in American units (feet and inches). When METRIC is specified  
**METRIC** FATAR will list tape lengths and error positions and lengths in metric units (meters and centimeters).  
Default: NONMETRIC

## 23.2 CONTINUED . . .

## 23.2.5 MISCELLANEOUS OPERANDS

**COMBFILES** Combine all files found on TAPEIN into a single file on TAPEOUT. The tape labels on TAPEOUT will show the DCB characteristics copied from the first file on TAPEIN; COMBFILES is intended for use when all the files on TAPEIN have similar characteristics. If the DCB characteristics of the TAPEIN files vary, various FATAR errors may occur, and any application which reads the combined output tape may also experience errors.

**EXIT=** Activate FATAR user open exit facility described in Section 21.4, output tape processing. The parameter "exitname" is the name of the load module to be loaded by FATAR and executed at open time for TAPEOUT.

Default: No exit is activated.

**EXPDAYS=** Used with VALIDATE=INEXP, documented below.

Default: 0

**LENCHK** If specified, FATAR will compare the length of every block read to the length of the preceding block and print the contents of the block if the lengths do not match. The first block in every file will always print. This can be used to scan a fixed length file for improper length blocks.

**RETCODE** RETCODE causes FATAR to terminate with a return code (condition code) of 12 rather than a U0200 or U0888 abend when serious errors are found.

**VALIDATE=** If LABEL=(,BLP) is specified on TAPEIN or TAPEOUT DD Statements, data management will not VERIFY the volume serial or expiration date of the tapes. However, if VALIDATE= is specified, FATAR will verify the volume serial of TAPEIN (VALIDATE=INPUT), TAPEOUT (VALIDATE=OUTPUT) or both (VALIDATE=ALL). The volume serial compared is the serial in the DD statement. For TAPEOUT, the expiration date in the tape label will be checked. If the wrong volume is mounted or if the expiration date is not yet reached, FATAR will issue message FATSW08 to the system operator giving the option of ignoring the error, mounting another tape, or terminating. TAPEOUT validation applies only to the first output volume.

In addition, VALIDATE=INEXP will validate the input volume serial and will also validate the expiration date of the input tape. If the input tape is expired, message FATSW08 will be issued allowing the operator to bypass copying tapes which have expired. If the EXPDAYS=nn operand is also specified, FATSW08 will be issued for expired input tapes and those which will expire in the next "nn" days, allowing you to bypass copying tapes which will expire shortly.

Default: No validation

**VERIFY** If VERIFY is specified and an output tape is being produced, at the conclusion of the copy FATAR will reopen the output tape as input and re-read it to verify its readability and contents (it is **not** compared to the input tape). All FATAR options are reset to their defaults and all MODIFICATION and SCAN Control Statements are ignored during the verification.

## 23.3 MODIFICATION/SCAN STATEMENTS

**MODIFICATION  
& SCAN  
STATEMENTS**

The FATAR block MODIFICATION and SCAN Control Statements (PRINT, DROP, KEEP, REPLACE, and SCAN) invoke optional facilities of FATAR to print data from TAPEIN, to scan TAPEIN data blocks for certain data or types of data, and to modify TAPEIN data blocks before they are written to TAPEOUT. The RENAME statement is also a MODIFICATION statement, but is used to rename files as they are copied to the output tape.

There may be as many MODIFICATION/SCAN statements as required. More than one statement may refer to the same block on TAPEIN; if so, the statements are executed in the order in which they appear in the control statement input. An exception to this is the PRINT Statement, since each data block will be printed only once; the **last** PRINT Statement which applies to a given data block is the one which will control its printing.

All operands must be specified on the MODIFICATION/SCAN Control Statements in the order shown in the statement formats below. However, it is not necessary to indicate the absence of optional operands with a comma. Because many of the operands are common on these statement, the operands are described separately in Section 23.4, except where special notes apply to a particular statement.

Use of these statements can be confusion. The FATAR examples in Section 26 contain examples of the use of each kind of MODIFICATION/SCAN statement. You may wish to review them as you read the following descriptions.

## 23.3.1 PRINT STATEMENT

**PRINT  
STATEMENT****PRINT****ALL**

**,LF=ALL|nH|n|nT**

**,B=ALL|b<sub>1</sub>|b<sub>1</sub>-b<sub>2</sub>**

**,L=1**

**,CHAR|HEX|DUMP**

The PRINT statement requests that data blocks from TAPEIN be printed after other MODIFICATION Statements have been applied. By default, the length and format of the printout are controlled by the PRTLEN=, CHAR, HEX, and DUMP operands on the ANALYZE/COPY Statement (which themselves default to 80 bytes in character format), but this can be overridden by the L= and CHAR/HEX/DUMP operands on the PRINT statement itself.

Note that the print length refers to the number of characters or bytes to print from each block, even if the blocks are being deblocked into logical records. If the length to print is longer than one logical record, the printout will be formatted into logical records.

The PRINT ALL statement causes all blocks in all files on TAPEIN to be printed, and cannot have any other operands on it. In this case, the length and format are controlled by the above-mentioned ANALYZE/COPY operands.

The ability to use the PRINT statement may be restricted by security rules; see Section 21.6.

**Examples**

```
PRINT ALL
PRINT LF=ALL , B=1 - 5
PRINT LF=3 , B=4 - 7 , L=400 , DUMP
PRINT LF=1T , B=ALL , HEX
```

## 23.3 CONTINUED . . .

## 23.3.2 DROP STATEMENT

**DROP STATEMENT**

**DCK**

,LF=ALL|nH|n|nT

,B=ALL|b<sub>1</sub>|b<sub>1</sub>-b<sub>2</sub>|b<sub>1</sub>.r<sub>1</sub>|b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub>

The DROP statement causes one or more blocks or records to be dropped (not copied from TAPEIN to TAPEOUT). If the DCK operand is specified, the drop will be done only if the affected block had a permanent data check when read from TAPEIN. If B=ALL is given, the entire file referenced by LF= will be dropped from TAPEOUT, including all tapemarks and labels (to drop all data from a file without dropping the file itself, specify "B=1-2147483647").

Dropping individual records (the "B=b<sub>1</sub>.r<sub>1</sub>" or "B=b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub>" formats), is effective only if the blocks on TAPEIN are being deblocked into records. If all of the records in a block are individually dropped, the entire block will be dropped.

**NOTE:** unless the KEEP operand is specified on the ANALYZE/COPY statement, the default operation of FATAR is to drop (not copy to TAPEOUT) any block that has a permanent data check on TAPEIN. It is not necessary to include a DROP DCK,LF=ALL,B=ALL statement to accomplish this. Including such a DROP statement will cause FATAR to unnecessarily generate a message for every block that it **did not drop**.

Examples:

```
DROP  LF=3 , B=ALL
DROP  LF=1 , B=1 - 2
DROP  LF=3 , B=5 . 6 - 6 . 2
DROP  DCK , LF=7 , B=ALL
```

## 23.3.3 KEEP STATEMENT

**KEEP STATEMENT**

**DCK**

,LF=ALL|nH|n|nT

,B=ALL|b<sub>1</sub>|b<sub>1</sub>-b<sub>2</sub>

,L=l

,P=p

,J=R

The KEEP statement causes blocks from TAPEIN to be copied to TAPEOUT, even if they would be dropped because of a DROP statement, or because of a permanent data check (the DCK operand is required to keep a data check block). Note that if you want to keep **all** such data check blocks, use the KEEP operand on the ANALYZE/COPY statement instead of a KEEP statement.

If the optional L= operand is given, the KEEP statement may be used to change the length of a block as it is being copied from TAPEIN to TAPEOUT. L= specifies the new block length and P=, if given, specifies location in the new block (relative to 1) where the old data will be placed. J=R requests that the old data be right-justified in the new block, with the rightmost byte of the old data block positioned at the location specified by P=; if J=R is omitted, the data is left justified, with the leftmost byte placed at the P= location. These combinations allow full control over repositioning of the data. Any extra bytes added at the front or end of the new block will be set to spaces (blanks). Whenever the length of a block is changed, deblocking of that block into logical records will be terminated.

Examples:

```
KEEP  LF=1 , B=ALL
KEEP  LF=ALL , B=1 - 5
KEEP  DCK , LF=1 , B=ALL , L=5000
```



## 23.3 CONTINUED . . .

## 23.3.4 REPLACE STATEMENT

<b>REPLACE STATEMENT</b>	<b>REPLACE</b>	<b>DCK</b> <b>,LF=ALL nH n nT</b> <b>,B=ALL b<sub>1</sub> b<sub>1</sub>-b<sub>2</sub> b<sub>1</sub>.r<sub>1</sub> b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub></b> <b>,P=p</b> <b>,V=string</b> <b>,S=string</b> <b>,D=string</b> <b>,ALL</b>
------------------------------	----------------	---

The REPLACE statement is used to modify data within blocks being copied from TAPEIN to TAPEOUT, and also to scan for blocks or records containing a specified data string (even if TAPEOUT is not present). If TAPEIN data blocks are being deblocked, the REPLACE Statement operates on a logical record basis.

If D= is specified without V= or S=, the data string specified by D= will be placed in every affected data record or block at the location given by P=.

If D= is specified with V=, the data in every affected data record or block will be verified for the current contents specified by V= before the data from D= is placed there (again, the location within the block or record is given by P=). This allows you to be sure the proper data field is being replaced, or to search a set of records for the proper one to modify.

If D= is specified with S=, every affected record or block will be scanned (starting at the location specified by P=, default 1) for the data string specified by S=. If found, the replacement data given by D= will be stored at the location where the S= data was found. Normally, only one such replacement will be done in each record or block, but if the parameter ALL is also given, the scan will continue until the end of the record/block.

If V= or S= is given without D=, verification or scanning will take place as described above, but no data modification will be done. This allows the REPLACE statement to be used to scan data for specific data strings, since any block on which REPLACE successfully operates will be eligible for printing.

Examples:

```

REPLACE  LF=ALL,B=ALL,S=C'ORIGINAL',D=C'NEW DATA',ALL
REPLACE  LF=3,B=ALL,P=10,V=X'1302',D=X'1503'
REPLACE  LF=ALL,B=5.2-20.6,S=C'FIND THIS STRING'

```

## 23.3 CONTINUED . . .

## 23.3.5 SCAN STATEMENT

<b>SCAN STATEMENT</b>	<b>SCAN</b>	<b>DCK</b> <b>,LF=ALL nH n nT</b> <b>,B=ALL b<sub>1</sub> b<sub>1</sub>-b<sub>2</sub> b<sub>1</sub>.r<sub>1</sub> b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub></b> <b>,L=l</b> <b>,P=p</b> <b>,T=t</b>
---------------------------	-------------	--

The SCAN statement causes specified data fields to be validated for specific type of content. Scans are performed on a logical record basis if records are being deblocked. The field to be scanned is at the location specified by P= (relative to 1) for the length specified by L=. If T=Z is specified, the field is checked for valid zoned decimal characters including decimal sign characters. If T=P is specified, the field is checked for valid packed decimal characters. For any other value of T=, there must be a corresponding TABLE Statement with the same T= value, and the SCAN statement will validate the presence or absence of the characters specified there. A message will be issued for every field which fails the SCAN test and the failing block will be printed.

Examples:

```
SCAN  LF=1 , B=ALL , L=5 , P=22 , T=P
SCAN  LF=ALL , B=ALL , L=7 , T=Z
SCAN  LF=3 , B=1-20 , L=5 , P=20 , T=A      ( see TABLE example )
```

## 23.3.6 TABLE STATEMENT

<b>TABLE STATEMENT</b>	<b>TABLE</b>	<b>T=t</b> <b>,R</b> <b>,D=string</b>
----------------------------	--------------	---

The TABLE statement is used to define a set of characters for SCAN statement operations. T= must specify a unique letter (except for P and Z which are reserved) for naming the table, so up to 24 tables can be defined. Each table can contain up to 256 characters, specified by the D= operand. The TABLE statement must precede any SCAN statement which references it. If a field being scanned contains any character not in the table, it will be flagged and its block printed. If the optional operand R is included, this operation is reversed, so that if the field contains any character which is in the table, it is flagged.

Examples:

```
TABLE T=A , D=C'0123456789ABCDEF' printable hex
TABLE T=A , D=C' ADFQR' ,
X'FF7F' , C'269'
```

## 23.3 CONTINUED . . .

## 23.3.7 RENAME STATEMENT

**RENAME  
STATEMENT****RENAME****UNCOND****,LF=ALL|n****,NEWN=newname|NEWI=newindexmask**

When an output tape is being created, RENAME can be used to change the name of the file(s) created on the output tape. The LF= operand specifies to which files on the input tape this RENAME statement applies; it can be for a single file, or for all files on the tape (in the latter case, the NEWI= operand is usually used).

NEWN= specifies a full replacement dataset name, up to 44 characters. If the name ends in a GDG relative generation number, e.g., NEWN=PROD.MAST(+1), a LOCATE done to get the proper absolute generation number. Note that if NEWN= is used with LF=ALL, all copied files on the output tape will have that name.

NEWI= specifies that the output dataset name is to be constructed by adding or replacing one or more index levels in the original name (from the input tape); replacement index levels do not have to be the same length as the original indexes they replace. In the simplest case, FATAR will use each index level specified in NEWI in place of the corresponding index in the original name. Any remaining index levels at the end of the name are copied unchanged. This can easily be used to change the first indexes of the name.

For example, if copying dataset A.B.C.D,

NEWI=D	results in	D . B . C . D	(first index replaced)
NEWI=DD . E	results in	DD . E . C . D	(first 2 indexes replaced)

If a period is specified without any preceding characters, FATAR will copy one original index level from the input dataset name to the output. This allows you to easily modify indexes in the middle of the name.

For example, if copying dataset A.B.C.D,

NEWI=. . E	results in	A . B . E . D	(third index replaced)
NEWI=FF . . . G	results in	FF . B . C . G	(first and fourth indexes replaced)

If + is specified before a new index level, FATAR will insert that new index into the output dataset name at that point. If ++ precedes the new index, it will be added to the end of the name. If - is specified, the next input index level will be dropped from (not copied to) the output name.

For example, if copying dataset A.B.C.D,

NEWI=+F	results in	F . A . B . C . D	(new first index added)
NEWI=. . +F	results in	A . B . F . C . E	(new third index added)
NEWI=++F	results in	A . B . C . D . F	(new last index added)
NEWI=. . +F	results in	A . B . F . C . D	(new third index added)
NEWI=. . -	results in	A . B . D	(third index dropped)
NEWI=Q . - . +E	results in	Q . C . E . D	(combination)

Note that, except for the ++ option, every period in the NEWI= mask corresponds to a period (one index level) in the original (input) dataset name. FATAR will check the resulting new name to insure it meets IBM standards.

**23.3 CONTINUED . . .**

If the NEWI= value ends in a GDG relative generation number, e.g., NEWI=..NEWMAST(-2), that relative number will be added to the end of the newname, and a LOCATE done to get the proper absolute generation number.

NEWI= is a convenient way to rename every file on a multi-file input tape as it is copied to the output tape, while using some index levels from the original name and replacing other indexes or adding new indexes.

Normally, the NEWI= operand is honored only if FATAR knows the full, accurate dataset name of the input file being copied; this is usually true only if the FATS/FATAR tape management interface is enabled (see Section 21.5). To use NEWI= when the only dataset name is the truncated 17-character name from the tape labels, specify the UNCOND operand; NEWI= will be applied to that truncated name except that if the truncated name starts with a period, it is discarded. If used, UNCOND must be the first operand on the RENAME statement.

The ability to use the RENAME statement may be restricted by security rules; see Section 21.6.

## 23.4 MODIFICATION/SCAN OPERANDS

The following are details of the keyword parameters which may appear on the MODIFICATION/SCAN Statements in the preceding sections. Consult the definitions of each statement to see which parameters are valid on each statement, and in what form.

**ALL** On a REPLACE statement containing the S= operand, specifies that all occurrences of the scan string in the current block or record are to be replaced. If omitted, only the first occurrence of the scan string is replaced.

On a PRINT statement, specifies that all data blocks are to be printed, using the formatting requested by the PRTLEN= and CHAR/HEX/DUMP operands on the ANALYZE/COPY statement.

**B=ALL** Specifies to which blocks within the file specified by LF= the statement applies.

**B=b<sub>1</sub>** **B=ALL** requests all blocks in the file.

**B=b<sub>1</sub>-b<sub>2</sub>** **B=b<sub>1</sub>** causes it to apply only to one specified block.

**B=b<sub>1</sub>.r<sub>1</sub>**

**B=b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub>** **B=b<sub>1</sub>-b<sub>2</sub>** specifies a range of blocks (b<sub>1</sub> less than b<sub>2</sub>).

**B=b<sub>1</sub>.r<sub>1</sub>** indicates a specific record within a specific block (record deblocking must be in effect).

**B=b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub>** causes the statement to apply to a range of records from record r<sub>1</sub> in block b<sub>1</sub> to record r<sub>2</sub> in block b<sub>2</sub> (b<sub>1</sub> and b<sub>2</sub> may be the same).

b<sub>1</sub>, b<sub>2</sub>, r<sub>1</sub>, and r<sub>2</sub> are all unsigned decimal numbers from 1 to 2,147,483,647 (without commas). Blocks within a file and records within a block are numbered starting with 1.

**CHAR** On a PRINT Statement, specifies an overriding print format (character, hexadecimal, or  
**HEX** dump) for this PRINT Statement only. If omitted, the print format specified on the  
**DUMP** ANALYZE/COPY Statement will control the format for this PRINT Statement.

**DCK** Specifies that FATAR is to perform the indicated operation only if the block affected had a permanent data check when read from TAPEIN.

**J=R** On KEEP Statements only, specifies that the old data block is to be right-justified in the new data block (if not specified, the old data block is left-justified).

**L=I** For PRINT, specifies an overriding print length.

For KEEP, specifies the new length of the block.

For SCAN, specifies the length of the field to verify.

I must be an unsigned decimal number.

**LF=ALL** Specifies the logical file on the input tape to which this statement applies. "n" is a  
**LF=n** decimal number from 1 to 10922 giving the logical file number. If the input tape is  
**LF=nH** labeled, this is the file sequence number, the same value you would specify in the  
**LF=nT** LABEL=n parameter in JCL. Specifying "n" alone will cause the statement to apply to the equivalent data file. Appending an "H" or "T" after the file number causes it to apply to the header or trailer label file, respectively, preceding or following that data file. If the input tape is not labeled, or if labels are not being processed (LABELS=NO), then "n" is the physical file number relative to the beginning of the tape (file 1).

LF=ALL specifies that this statement applies to all files of the input tape.

**NOTE:** For compatibility with Version 3.0 of FATAR, the parameter F=n will be accepted in place of LF=. However, in this case, "n" specifies the physical file number only regardless of the label status of the tape, and header and trailer label files must be counted. Use of F= is not recommended for new applications.

## 23.4 CONTINUED . . .

**P=p** For SCAN and REPLACE, specifies the starting location for the operation (if FATAR is deblocking logical records, specifies a location within a record). For KEEP, specifies the starting location (or ending location if J=R) of the old data block within the new block.

"p" must be an unsigned decimal number. P=1 is the default and specifies the first byte of the record or block. For deblocked variable format records, the first data byte is at location 5, past the 4 byte RDW (Record Descriptor Word).

**T=t** On a TABLE Statement, defines the name of a table for reference by a SCAN Statement. "t" must be a unique letter other than P and Z, which are reserved.

On a SCAN Statement, specifies the type of scan to be performed:

T=Z – scan for zoned decimal numbers

T=P – scan for packed decimal numbers

T=t – scan according to user table "t"

**V=string** Defines a data string on a REPLACE or TABLE statement. The data may be in EBCDIC characters or hexadecimal bytes, or a combination of both, using this format:

**S=string**

**D=string**

**C'chardata'** – defines an EBCDIC string up to 255 bytes in length.

**X'hexdata'** – defines a hexadecimal string up to 255 bytes in length.

A string consisting of both character and hexadecimal data may be defined by simply placing the C and X operands after one another, separated by commas. For example,

D=C'ABC',X'43C721',C'XYZ

defines a 9-byte string consisting of the 3 separate strings concatenated together.

Longer strings obviously cannot fit on a single control statement record, so a data string is continued onto another record by closing the string with a quote and comma followed by a blank, and starting on the next card (in any column) with X' or C'. For example,

V=C'THIS IS A VERY LONG STRING THAT IS GOING TO BE ',  
C'CONTINUED ON THE NEXT RECORD'

**24.0 FATAR SPECIAL CONSIDERATIONS**

This section will examine special considerations for the use of FATAR, primarily recovering from special sorts of errors.

**24.1 VARIABLE SPANNED RECORDS**

**S002 ABENDS** S002 abends on variable spanned format tapes (RECFM=VBS or VS) can be very frustrating because of the difficulty in diagnosing and correcting the errors. By far the most common use of VBS format is for SMF/RMF data, since their format is variable with few limitations on the maximum logical record size, so a S002 abend on those tapes may threaten an installation's accounting and performance data.

S002 abends usually indicate an improperly spanned logical record, due to some logical or physical error when creating the tape, or possibly due to blocks dropped when copying with FATAR because of permanent data checks. FATAR is one of the few programs which can detect and correct these spanning errors, but it is a process requiring several passes of the tape and some human analysis.

**VARIABLE  
SPANNED  
FORMAT**

Variable spanned tapes have the characteristic that logical records may "span" across two or more physical blocks. Because of this, a logical record may be larger than the blocksize of the file. Because VBS format attempts to fill each physical block to capacity, the last logical record in each block, regardless of size, will almost always span to the next block, so almost every block begins and ends with a spanned record.

When a record is broken into pieces for spanning, each piece is referred to as a segment. There are flags in the RDW (record descriptor word, preceding each variable length record) which indicate "first segment", "middle segment", or "last segment" of the total logical record ("middle segment" exists only when a logical record spans three or more blocks).

**Note:** IBM uses VBS format for SMF data on disk because the maximum SMF record size (32760) exceeds the blocksize typically used for the SMF datasets. However, when SMF data is copied to tape, there is no reason that it must remain in VBS format. Since tapes can easily support blocks up to 32767 bytes in length, Innovation recommends that SMF data be converted to RECFM=VB (not spanned) during the copy to tape so that spanning errors on SMF data can be avoided altogether. Many utilities that are used for the SMF data copy can do this conversion. Unfortunately, even on a tape-to-tape copy, FATAR cannot do this type of conversion.

**FATAR  
SPANNED  
FORMAT  
CHECKING**

FATAR automatically performs spanning checks on VS or VBS tapes, detecting missing first or last segments. When FATAR detects a missing segment, it prints the message FATS069 START (or END) OF SEGMENT MISSING. The block and record numbers printed to indicate where the error was detected, but the actual problem is usually in the preceding block. End of segment missing indicates that FATAR found the start of a new logical record before finding the end of the preceding logical record, which started in one of the immediately preceding blocks.

Start of segment missing indicates that FATAR found a middle or end segment not preceded by a start segment, which again should have been in the preceding block. One situation FATAR cannot detect is a missing middle segment where the start and end segments exist; it might even be that those start and end segments do not even belong together because of lost intermediate blocks, but there is no way to detect that. These errors will not usually cause S002 abends, but may cause data errors in the application programs reading them. If you can identify the records causing the error, FATAR may still be used to correct this problem.

## 24.1 CONTINUED . . .

**CORRECTING  
S002 ABENDS**

To correct these problems with FATAR, follow this procedure:

- 1) Analyze the tape once with FATAR (no TAPEOUT) and note which blocks and records FATAR identifies as having spanning errors (FATS069 messages).
- 2) Analyze the tape again, printing the complete blocks for at least three blocks before and after each block identified by FATS069, using this input:

```
ANALYZE PRTLEN=32760,DUMP  
PRINT  LF=f,B=b1-b2
```

substituting the proper file number (f) and a range of block numbers (b1-b2). Include multiple PRINT statements if multiple FATS069 messages occurred.

- 3) Examine the printout to see where various logical records start and end. FATAR will print each physical block broken down into logical records; it prints the block number and then numbers the logical records (or segments) in the block. You can see the spanning flags in the printout in byte 3 of each segment:

00 – complete logical record (not spanned)

01 – start segment

02 – end segment

03 – middle segment

- 4) If you received START OF SEGMENT MISSING, then the middle and end segments left must be deleted. Starting at (and including) the block and record number indicated in the FATS069 message, look at each segment until you find the end segment (byte 3 = x '02'). Write down the block number and record number of each segment you examine (including the end segment) since each must be deleted.
- 5) If you received END OF SEGMENT MISSING, then start at the record preceding the one identified in the FATS069 message and look at each record going backwards until you find the start segment (byte 3 = x '01'). Write down the block and record number of each segment examined in order to delete them. Do not include the record from the FATS069 message.
- 6) Run FATAR again with a TAPEOUT DD statement in order to copy and correct the spanned tape, using this input:

```
COPY VERIFY  
DROP  LF=f,B=b.r
```

substituting the proper file number (f), block number (b) and record number (r) to be deleted. Include a DROP statement for each segment you identified above. The dropped segments should eliminate the spanning errors, and the verify option will cause the output tape to be analyzed to be sure (no FATS069 messages should occur during the verification phase).



**24.2 RECOVERING DATA FROM OVERWRITTEN TAPES****OVERWRITTEN  
TAPES**

A special recovery problem exists when a tape which still contains important data is accidentally used for output, resulting in an overwritten tape. Although tape management systems usually prevent accidentally overwriting of a tape which is not in scratch status, sometimes a tape will expire and be used for output while the original data is still needed, probably because of incorrect assignment of tape expiration dates.

Any data that is physically overwritten is gone, unfortunately, but if the new data file is shorter than the original data file, it is possible to recover the remains of the original data and copy it to another tape. This recovered data may or may not be useful, depending on the format of the data and its intended use.

To recover the remains of the partially overwritten data file, you must position FATAR to the beginning of that data, allowing FATAR to read and copy the remaining data. Since the header labels of the file are also overlaid, you will need some knowledge of the original data to be able to specify proper DCB information. Examples in Section 26.2 shows how you might do this.

Because the first block of the remaining original data is probably partially overwritten, you will almost always get a data check on that block. Since FATAR will automatically discard that block, this can be ignored; the remaining blocks should be readable. Occasionally, the error on that first block will make the tape drive believe that a more serious error exists and refuse to read further. If this reoccurs on several different tape drives, the tape is probably not recoverable.

If the tape originally contained multiple data files, it may be that some of the original files are intact with complete header and trailer labels, yet they cannot be opened normally because of the overlaying data earlier on the tape. You can use FATAR to completely and automatically copy those files, but you need to position FATAR to the header labels of the first such file with LABEL=(n,BLP). Examples in Section 26.2 shows an example of this function. The tape layouts below will help you understand the value of "n" to be used, but it may require some experimentation to get the right value.

**EOD MARKS**

On tapes written on a cartridge drives (except 3480) the drive writes an EOD (End-of-Data) mark after the last written data on each tape volume. The EOD is designed to make it more difficult for an end-user to read past the end of a data file to which he is authorized and read residual unauthorized data from the tape. But it makes the job of a legitimate recovery more difficult. When the tape is overwritten, an EOD is written immediately after the overwriting file. Normal access methods cannot get past the EOD.

Using techniques which Innovation has obtained from IBM and other vendors, FATAR will attempt to bypass the EOD mark and recover the data beyond it. The technique used may depend on the manufacturer of the drive. In some cases the recovery may not be successful. In others, it may work but is forced to skip over some of the original data before beginning the recovery.

The ability of FATAR to read data past an EOD mark may be restricted by security rules; see Section 21.6.

## EXAMPLES OF OVERLAYING DATA

If the tape originally contained a single data file, it looked like this:

[illegible]

V O L 1	H D R 1	H D R 2	T M	DATA2	T M	E O F 1	E O F 2	T M	T M	E O D	DATA1 remainder	E O F 1	E O F 2	T M	T M	E O D
------------------	------------------	------------------	--------	-------	--------	------------------	------------------	--------	--------	-------------	--------------------	------------------	------------------	--------	--------	-------------

V O L 1	H D R 1	H D R 2	T M	DATA2	T M	E O F 1	E O F 2	T M	H D R 1	H D R 2	T M	DATA3	T M	E O F 1	E O F 2	T M	T M	E O D	DATA1 remainder	T M	E O F 1	E O F 2	T M	T M	E O D
------------------	------------------	------------------	--------	-------	--------	------------------	------------------	--------	------------------	------------------	--------	-------	--------	------------------	------------------	--------	--------	-------------	--------------------	--------	------------------	------------------	--------	--------	-------------

V O L 1	H D R 1	H D R 2	T M	DATA1	T M	E O F 1	E O F 2	T M	H D R 1	H D R 2	T M	DATA2	T M	E O F 1	E O F 2	T M	T M	E O D
------------------	------------------	------------------	--------	-------	--------	------------------	------------------	--------	------------------	------------------	--------	-------	--------	------------------	------------------	--------	--------	-------------

V O L 1	H D R 1	H D R 2	T M	DATA3	T M	E O F 1	E O F 2	T M	T M	E O D	DATA1 remainder	T M	T M	E O F 1	E O F 2	T M	H D R 1	H D R 2	T M	DATA2	T M	E O F 1	E O F 2	T M	T M	E O D
------------------	------------------	------------------	--------	-------	--------	------------------	------------------	--------	--------	-------------	--------------------	--------	--------	------------------	------------------	--------	------------------	------------------	--------	-------	--------	------------------	------------------	--------	--------	-------------

- 24-4 -

## 24.2 CONTINUED . . .

**RECOVERY  
TECHNIQUES**

- 1) If you are recovering from an overwritten tape that supports EOD marks (all cartridge drives except 3480s and compatibles), recovery is fairly easy. Specify LABELS=EOD on the ANALYZE/COPY statement and FATAR will do most of the work of recovering all the residual data on the tape.

LABELS=EOD instructs FATAR to read up the tape until an EOD mark is encountered. Then it reads the data following that EOD mark. The first file after the EOD mark is (usually) partially overwritten, so you have to specify a dataset name and DCB information on the TAPEOUT DD statement so that copied data can be used. If there are additional files on the tape beyond that first overwritten file, they will also be copied, using the dataset names and DCB information in the header labels of those files. No special LABEL= parameters are required on TAPEIN but TAPEOUT should specify or default to LABEL=(,SL). See example 22 in Section 26.2

- 2) For round tapes and 3480 cartridges, recovery is a 2 step process:

- a) First, recover the partially overwritten data file. Specify LABELS=NO on the ANALYZE/COPY statement and LABEL=(n,BLP) on the TAPEIN DD to position FATAR to the start of that file, where "n" is the physical file number of that data. You can calculate "n" by counting the number of tape marks preceding that data and adding 1. You can see in the example tape layouts on the previous page that "n" is usually 5 if there is only one overwriting file. See example 23 in Section 26.2.
- b) If there were additional files on the tape beyond the partially overwritten file, you recover them by specifying LABELS=YES on the ANALYZE/COPY statement and LABEL=(n,BLP) on the TAPEIN DD to position FATAR to the header labels of the first additional file, where "n" is the physical file number of those header files. "n" is usually 7 if there is only one overwriting file. See example 24 in Section 26.2.
- c) If you are unsure of the format or size of the original or overlaying files, you can run a FATAR ANALYZE/COPY on the tape with NUMFILES=999 and LABELS=NO; this will continue reading past the 2 consecutive tape marks at the end of the overlaying data file and show you what remains on the tape.

## 24.8 FATAR REPORT EXAMPLES

In this example, FATAR was executed with no control statements (all defaults). The input was a standard label 3480 cartridge containing 2 files; device type 3480XF is a drive capable of IDRC compaction, but this tape is not compacted. There is a data check on a block in the middle of the second file. 80 bytes of the block with the data check are displayed, as well as for the block before and after the data check block.

FATS014 NO CONTROL CARD DATA SET. DEFAULTS ASSUMED  
FATS071 TAPE BUFFER SIZE IS 65535 BYTES

## CHARACTERISTICS OF THE TAPE TO BE ANALYZED

```

UNIT DEVICE SERIAL
0391 3480XF 800432
FATAR DETAIL REPORT
1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
(COLUMN GRID IS VALID ONLY FOR CHARACTER FORMATTED DATA)

BLOCK LENGTH/ MESSAGE/
NUMBER DISPL BLOCK TYPE
* * * * *
1 80 INPUT LABEL 1H
***VOLUME LABEL***
2 80 INPUT LABEL
***HEADER LABEL***
3 80 INPUT LABEL
***HEADER LABEL***
* * * * *
END OF FILE 1H -- FILE CONTAINED 3 BLOCKS
* * * * *
START FILE 1
* * * * *
END OF FILE 1 -- FILE CONTAINED 486 BLOCKS
* * * * *
START FILE 1T
1 80 INPUT LABEL
***END-OF-FILE***
2 80 INPUT LABEL
***END-OF-FILE***
* * * * *
END OF FILE 1T -- FILE CONTAINED 2 BLOCKS
* * * * *
START FILE 2H
1 80 INPUT LABEL
***HEADER LABEL***
2 80 INPUT LABEL
***HEADER LABEL***
* * * * *
END OF FILE 2H -- FILE CONTAINED 2 BLOCKS
* * * * *
START FILE 2
8 17718 ***PERMANENT DATA CHECK -- RETRIES= 0, CLEANING ACTIONS= 0, ROR RETRIES= 0
DATA TRANSFERRED : : : : : : : : : : : : : : : :
BLOCK DROPPED
7 47626 BLOCK PRECEEDING D/C : : : : : : : : : : : : : : : : : : : : : :
9 44970 BLOCK FOLLOWING D/C : : : : : : : : : : : : : : : : : : : : : :
* * * * *
END OF FILE 2 -- FILE CONTAINED 580 BLOCKS
* * * * *
START FILE 2T
1 80 INPUT LABEL
***END-OF-FILE***
2 80 INPUT LABEL
***END-OF-FILE***
* * * * *
END OF FILE 2T -- FILE CONTAINED 2 BLOCKS
* * * * *
START FILE 3H
* * * * *
END OF FILE 3H -- FILE CONTAINED 0 BLOCKS
FATS020 ANALYSIS TERMINATED AT TAPemark SEQUENCE

FINAL TOTALS
FILES READ BLOCKS READ BYTES READ FEET READ TEMP ERRS PERM ERRS FILES WRITTEN BLOCKS WRITTEN
2 1066 40367286 96 0 0 0 0

```

FATAR		TAPE SUMMARY FOR TAPE VOLUME -800432- AT DENSITY 38000 BPI										06/01/2001			
PHYS DATA SET NAME	FILE	FIL#	REC-	LRECL	CREATING	BLOCKS	BYTES	PERM	---BLOCKSIZES---			EST.			
FILE (LAST 17 CHARS)	SERIAL	VOL#	CRDATE	EXPDATE	FM	BLKSZ	JOB&STEP	SEC	READ	TEMP	MIN	AVG	MAX	FEET	
2 RAY.SCR010	800432	0001	1986/192	00/000	U	00000	SPSN	0	486	17M	0	24	34213	48092	39
		0001				32760	COPY1				0				
5 RAY.SCR501	800432	0002	1986/192	00/000	U	00000	SPSM	0	580	24M	0	24	40931	49060	56
		0001				32760	COPY1				0				
HIGHEST EXPIRATION =====>			00/000		TOTALS =====>			1066		40M		0		95	



**24.8 CONTINUED . . .**

Because the data on this input tape was compacted (compressed) by the tape hardware, FATAR gives you several statistics about the compaction. Compaction (called IDRC on some types of cartridge) not only compressed the data, it combines user blocks into larger "superblocks" on the tape, all transparent to the user. In the detail report, the final totals show the number of blocks and bytes written by the user, and also shows the physical blocks and bytes written after compaction. The percentage of tape saved (length of tape required for the uncompact data vs the length actually used) is displayed. Since the uncompact data is not written to tape, the actual length (in meters in this example) is displayed only for the compacted data.

In the summary report, the same information is displayed, plus compaction statistics for each input file.

**26.0 FATAR EXAMPLES OF USAGE****INTRODUCTION**

Examples are provided here to guide you in the use of FATAR. As many of the common uses of FATAR as possible have been included, and most of the control statements and keywords are illustrated here. However, be aware that they are just examples and must be customized to your installation and intended use. Areas of customization will probably include: dataset names, unit names for tape, special parameters for tape management systems. The examples also assume that no special STEPLIB DD Statement is required to execute FATAR; this may not be true in your installation.

Because of the many possible combinations of options, control statements, tape label types, etc., it is not practical to create examples which cover all possible FATAR uses. Many examples illustrate more than one aspect of FATAR use, e.g., a particular tape label type and use of some FATAR keyword. This does not mean that the two must always go together unless the accompanying text says so.

Some examples use LABEL=(,BLP) on the input tape in order to be able to mount a labeled tape without knowing its volume serial or dataset names. Sometimes this is simply a convenience, but for some operations, such as printing the labels from an unknown tape, it is essential. In some installations, system parameters prevent the usage of BLP by most users; in this case, the FATAR option BLP on the ANALYZE/COPY Statement can be substituted. See Section 21.6 for security considerations for the use of BLP.

Some examples illustrate the use of the CAT= operand to catalog output datasets and the RENAME statement with NEWI= to rename copied datasets. These may not work as shown unless you have the FATAR tape management interface (Section 21.5) enabled, to allow FATAR to get the full 44-character dataset name of the input dataset.

**ISPF DIALOG  
SUPPORT**

**Most of the FATAR functions can be performed using the FATAR ISPF Interface. See Section 93 for further details.**

## 26.1 INDEX TO EXAMPLES

**INDEX TO  
EXAMPLES**

The first-time or infrequent user of FATAR should review the examples below in order since they frequently build upon one another in illustrating the usage and effects of various parameters. For your convenience, however, here is a list of the examples provided to aid in quickly finding the one you need. It shows the title of the example, the ANALYZE/COPY Statement keywords it illustrates, and the other Control Statements it illustrates.

Note that the analyze and copy examples will properly handle tapes with block sizes up to 256K (262143). Tapes with block sizes over 32K (32760) can be created only by applications running on OS/390 2.10 or z/OS (except that backup products such as FDR and DFSMSdss will create tapes with blocks up to 64K even on older systems).

#	<u>EXAMPLE NAME</u>	<u>KEYWORDS</u>	<u>STATEMENTS</u>
1	Analyze any tape		
2	Analyze an SL tape	LENCHK,LBLPRT	PRINT
3	Analyze a partial SL tape	NUMFILES,PRTLEN	PRINT ALL
4	Copy an SL tape	CAT	
5	Copy any tape to an SL tape	VERIFY,KEEP	DROP
6	Partial copy of an SL tape	NUMFILES,REBLOCK, COMPACTION,CAT	RENAME
7	Partial copy of an SL tape in BLP mode		PRINT
8	Analyze a multi-volume tape set		PRINT,SCAN
9	Analyze an SL tape with no label processing	LABELS	
10	Analyze an NL tape	NF,PRTLEN PRINT,SCAN,TABLE	
11	Search a partial file	ENDAFTER, PRTLEN,LBLPRT	REPLACE
12	Copy an ANSI tape		
13	Copy an NL tape	NUMFILES,VERIFY	KEEP
14	Copy an SL file to NL	NUMFILES	
15	Image copy a tape	LABELS	
16	Copy a multi-file, multi-volume to 3590	VERIFY,CAT	RENAME
17	Copy and modify a tape file	MSGLEVEL,CAT	REPLACE
18	Copy and modify a tape file	CAT	REPLACE
19	Recover from data checks	RETRY,KEEP	REPLACE
20	Recover from data checks on multi-volume tape		
21	Recover from data checks	LABELS	
22	Recover data from an overwritten tape	NF,CAT	
23	Copy files from an overwritten tape	CAT	RENAME
24	Copy an improperly closed tape	ENDAFTER	
25	Exclude files from a multi-file copy	VERIFY	DROP
26	Copy user labels		
27	Drop an extra tape mark	NUMFILES,CAT	DROP
28	FATAR as a FATS sub-task	VOL,CAT	
29	FATAR as a FATS sub-task	VOL,VOLINCR,MAXVOLN, VERIFY	



## 26.2 EXAMPLES

**EXAMPLE 1:  
ANALYZE ANY  
TAPE**

Analyze any tape. Since LABEL=(,BLP) is specified on the TAPEIN DD Statement, any tape can be mounted when a mount for volser 999999 is issued; the DSN= in the JCL does not have to be correct. You may need to specify EXPDT=98000 (as shown) so that your tape management system does not attempt to verify volume 999999. The tape may have any type of labels (IBM standard, ANSI, or none at all); FATAR will automatically determine the label type and handle them if present. If the labels are ANSI, both labels and data will be translated to EBCDIC. FATAR will print a summary of the files and data on the tape on the optional TAPESUMM DD Statement. FATAR will print all tape labels, count the blocks in every file, and identify any block causing a temporary or permanent data check, printing the first 80 characters of that block as well as the blocks preceding and following it. All blocks will be deblocked into logical records (if the tape is labeled). All files on the tape will be processed.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=999999,
//           LABEL=(,BLP,EXPDT=98000),DISP=OLD
//SYSIN      DD     *
              ANALYZE
```

**EXAMPLE 2:  
ANALYZE AN  
SL TAPE**

Analyze standard label tape 111111. Since it is being opened as a labeled tape (no LABEL parameter on the TAPEIN DD Statement), the volume serial and first dataset name must be specified correctly; your tape management system may verify these values. The parameter LENCHK causes FATAR to print the beginning of any block which differs in length from the preceding block (for checking fixed length files). The parameter LBLPRT=DUMP causes labels to be printed in dump format (hex and EBCDIC). The PRINT statement causes blocks 1 to 3 of every data file to be printed. All other operations are the same as Example 1 above.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=111111,
//           DISP=OLD
//SYSIN      DD     *
              ANALYZE LENCHK,LBLPRT=DUMP
              PRINT LF=ALL,B=1-3
```

**EXAMPLE 3:  
ANALYZE A  
PARTIAL SL  
TAPE**

Analyze the fourth and fifth files on standard labeled tape 111111, where the fourth file is called "FOURTHDS"; your tape management system may verify these values. The LABEL=4 parameter on the TAPEIN DD Statement tells FATAR to begin with that dataset, and the NUMFILES=2 parameter on the ANALYZE Statement limits the number of files FATAR will analyze. The PRINT ALL Control Statement directs FATAR to print all data blocks (first 200 bytes).

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FOURTHDS,UNIT=TAPE,VOL=SER=111111,
//           LABEL=4,DISP=OLD
//SYSIN      DD     *
              ANALYZE NUMFILES=2,PRTLEN=200
              PRINT ALL
```

## 26.2 CONTINUED . . .

**EXAMPLE 4:** Standard labeled tape 111111 will be copied to standard labeled tape 222222. All files will be copied. Each file on tape 222222 will receive the dataset name of the original file on tape 111111; the dataset name on the TAPEOUT DD statement will not be used. If any blocks have unrecoverable data checks, they will be dropped from the output tape, and the block counts in the dataset trailer labels adjusted. If you have the FATAR tape management interface enabled, it will be used to get the full 44-character dataset names of every input file; if this is successful, CAT=RECAT will cause every output dataset to be recataloged to the output tape.

```
//FATAR      EXEC  PGM=FATAR,REGION=0M
//SYSPRINT   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEIN     DD    DSN=FIRSTDS,UNIT=TAPE,VOL=SER=111111,
//           DISP=OLD
//TAPEOUT    DD    DSN=NOT.USED,UNIT=TAPE,VOL=SER=222222,
//           DISP=(NEW,KEEP)
//SYSIN      DD    *
COPY        CAT=RECAT
```

**EXAMPLE 5:** Tape 999999 will be copied to a scratch standard label volume. All files will be copied. FATAR will determine the label type of the input tape (STANDARD, ANSI, or UNLabeled) and process it appropriately. If the input labels are ANSI, all data files will be translated from ASCII to EBCDIC (if the records are ANSI variable length, RECFM=D, they will be converted to IBM RECFM=V). The dataset names from the input tape will be copied to the output (if TAPEIN is unlabeled, the DSN= value from the TAPEIN DD statement is used to name all output files); if the FATAR tape management interface is enabled, it will be used to get the full 44-character dataset name of every input file. The volume serial of the output tape will be preserved. If the output tape is filled before the input data is exhausted another volume will be requested. If any blocks on the input have unrecoverable data checks, they will be identified and printed, but the KEEP parameter will cause them to be written to the output tape anyway (using whatever data the input tape was able to transfer). The DROP statements will prevent the indicated input blocks from being written to the output tape. The VERIFY parameter will cause the output tapes to be analyzed after the copy is complete to verify readability.

```
//FATAR      EXEC  PGM=FATAR,REGION=0M
//SYSPRINT   DD    SYSOUT=*
//TAPESUMM   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEIN     DD    DSN=FIRSTDS,UNIT=TAPE,VOL=SER=999999,
//           LABEL=(,BLP),DISP=OLD
//TAPEOUT    DD    DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD    *
COPY        KEEP,VERIFY
DROP  LF=3,B=2
DROP  LF=3,B=10-12
```

## 26.2 CONTINUED . . .

**EXAMPLE 6:  
PARTIAL COPY  
OF AN SL TAPE**

Files 3 and 4 will be copied from labeled tape 111111 and written to a scratch Magstar (3590) tape. The LABEL=3 parameter on the TAPEIN DD Statement causes FATAR to start the copy with file 3 of the tape; the dataset name of that file must be properly specified. The NUMFILES=2 parameter tells FATAR to copy two logical files, namely files 3 and 4. The files will become files 1 and 2 on the output tape, with the dataset names of the original files. COMPACTION=COMP will cause the output tape to be written using IDRC hardware compaction. REBLOCK=16000 will cause each copied file to be reblocked to as close to a blocksize of 16000 bytes as possible. The RENAME statement is used to specify new names for the 2 output files, and CAT=YES will cause FATAR to catalog those names to the output tape.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=THIRDDS,UNIT=TAPE,VOL=SER=111111,
//            LABEL=3,DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=3590-1,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY         NUMFILES=2,REBLOCK=16000,COMPACTION=COMP,CAT=YES
RENAME  LF=3,NEWN=TECHSERV.FILE3
RENAME  LF=4,NEWN=TECHSERV.FILE4
```

**EXAMPLE 7:  
PARTIAL COPY  
OF AN SL TAPE**

All files on labeled tape 111111 from file 4 through the end of the tape are copied to labeled output tape 222222, starting at file 2. In other words, file 4 will be copied to file 2, file 5 will be copied to file 3, etc. Dataset names from the input tape will be retained; if the FATAR tape management interface is enabled, it will be used to get the full 44-character dataset name of every input file.

DCB=RECFM=U on the input tape will cause FATAR to treat the blocks as unblocked for printing/modification purposes but will not affect the DCB which will be written in the output tape's labels (which will be copied from the input tape's labels). LABEL=(10,BLP) is required on TAPEIN since we do not know its dataset name and since FATAR must be positioned to the header labels of the first file. The rule is "SL file number, times 3, minus 2", in this case  $4*3-2=10$ .

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=UNKNOWN,UNIT=TAPE,VOL=SER=111111,
//            LABEL=(10,BLP),DISP=OLD,DCB=RECFM=U
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,VOL=SER=222222,
//            LABEL=2,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY
PRINT ALL
```

## 26.2 CONTINUED . . .

**EXAMPLE 8:  
ANALYZE A  
MULTI-VOLUME  
TAPE SET**

Analyze multi-volume labeled tape set 333333, 444444, and 555555, which contain more than one dataset. FATAR will properly read all three volumes, giving total block counts for each dataset even when it crosses volumes (an intermediate block count is given each time an end-of-volume is reached). The tape summary (DD statement TAPESUMM) will contain one page for each tape, detailing the files and block counts on it. File numbers will be logical files since the beginning of the first tape, counting files which cross multiple volumes as one file. The PRINT control statements instruct FATAR to PRINT block 7 of file 1 in hexadecimal, blocks 22 and 23 of file 2 in dump format (HEX and EBCDIC) printing 96 bytes, and block 125 of file 4 in EBCDIC printing 250 bytes. The SCAN control statement requests FATAR to verify that the first 4 bytes of every record in file 2 contains a valid zoned decimal number.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=UNKNOWN,UNIT=TAPE,
//            VOL=SER=(333333,444444,555555),
//            LABEL=(,BLP,EXPDT=98000),DISP=OLD
//SYSIN      DD     *
ANALYZE
PRINT  LF=1,B=7,HEX
PRINT  LF=2,B=22-23,L=96,DUMP
PRINT  LF=4,B=125,L=250
SCAN   LF=2,B=ALL,L=4,P=1,T=Z
```

**EXAMPLE 9:  
ANALYZE AN  
SL TAPE WITH  
NO LABEL  
PROCESSING**

Analyze SL tape 111111 without label processing, so that labels will be processed as data (however, labels will still be formatted and printed). Processing will continue until two consecutive tape marks are read or until EOVL labels are read (tapes containing datasets which overflowed to another tape and end with EOVL labels will not end with two tape marks, so FATAR has a special SCAN for the EOVL labels). Since labels are not processed and no DCB information appears in the JCL, FATAR will process all tape blocks as RECFM=U (unblocked). Since FATAR is not processing labels, logical file numbers refer to the physical file number relative to the beginning of the tape, and the header and trailer label files are counted.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=111111,
//            DISP=OLD
//SYSIN      DD     *
ANALYZE LABELS=NO
```

**EXAMPLE 10:  
ANALYZE AN  
NL TAPE**

Analyze unlabeled tape 666666 with deblocking. The NF=3 parameter limits FATAR to reading three physical files. Since DCB information is supplied on the TAPEIN DD statement, FATAR will attempt to deblock every file processed. On the PRINT and SCAN statements, the LF= parameter refers to the physical file number since the beginning of the tape. The SCAN statement is used to VERIFY that bytes 4 to 6 of all records from record 5 in block 3 to record 2 in block 5 contain only characters A, B, or C (as defined in the TABLE statement). The PRTLEN=5000 will ensure that any blocks printed (due to the PRINT statement or failure of the SCAN statement) will print the entire block (up to 5000 bytes), deblocked into records.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=NOLABEL,UNIT=TAPE,VOL=SER=666666,
//            DCB=(RECFM=FB,LRECL=120),
//            LABEL=(,NL),DISP=OLD
//SYSIN      DD     *
ANALYZE NF=3,PRTLEN=5000
PRINT  LF=1,B=1-20
TABLE  T=A,D=C'ABC'
SCAN   LF=2,B=3.5-5.2,L=3,P=4,T=A
```

## 26.2 CONTINUED . . .

**EXAMPLE 11:** Analyze a partial file on SL tape 111111 and print blocks containing a certain string. Since only the first 20 blocks are to be scanned, there is no need to read the rest of the tape; the ENDAFTER parameter causes FATAR to terminate immediately after processing block 20 in file 1 (since labels are being processed, this is the first data file). The REPLACE statement tells FATAR to print any block containing the string "EQUIPMENT" in any position, and the PRTLEN=32767 parameter causes the entire data block to be printed. The LBLPRT=HEX parameter causes the header labels of the file to be printed in hexadecimal.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=UNKNOWN,UNIT=TAPE,VOL=SER=111111,
//           LABEL=(,BLP,EXPDT=98000),DISP=OLD
//SYSIN      DD     *
ANALYZE ENDAFTER=(LF=1,B=20),PRTLEN=32767,LBLPRT=HEX
REPLACE LF=1,B=1-20,S=C'EQUIPMENT'
```

**EXAMPLE 12:** ANSI labeled tape 333333 is to be copied to a scratch tape with ANSI labels. The TAPEIN DD statement specifies LABEL=(,AL), but it could also specify LABEL=(,BLP) if the true volume serial or first dataset name were not known; FATAR works correctly in either case. The TAPEOUT DD statement must specify LABEL=(,AL) for the labels to be written in ANSI format and the data in ASCII character set. All other processing is just the same as for standard label tapes (see example 4).

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=333333,
//           LABEL=(,AL),DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,
//           LABEL=(,AL),DISP=(NEW,KEEP)
//SYSIN      DD     *
ANALYZE
```

**EXAMPLE 13:** 7 files from unlabeled tape 222222 are to be copied to a scratch unlabeled tape. NUMFILES=7 tells FATAR to copy 7 files (7 tape marks and all data preceding them). Note that NUMFILES=0 could have been specified or defaulted to cause FATAR to copy all files until two consecutive tape marks were read. If the FATAR tape management interface is enabled, it will be used to get the true dataset name of every input file and use it to name the equivalent output file. The first KEEP statement causes block 7 of file 2 to be written to the output tape even though it is known to cause a data check. The second KEEP statement causes block 12 of file 3 to be increased in length to 1400 bytes, with the original data block positioned at byte 51 (50 bytes of spaces will be added to the start of the block; if the original block is less than 1350 bytes long, additional spaces will be added to the end). The VERIFY parameter will cause the output tape to be analyzed for readability; however, if the output tape requires multiple volumes, the verify will be suppressed (multi-volume NL tapes cannot be verified).

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=222222,
//           LABEL=(,NL),DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,
//           LABEL=(,NL),DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        NUMFILES=7,VERIFY
KEEP DCK,LF=2,B=7
KEEP LF=3,B=12,L=1400,P=51
```

## 26.2 CONTINUED . . .

**EXAMPLE 14:** File 3 from standard label tape 111111 is to be copied to an unlabeled scratch tape. LABEL=3 on the TAPEIN DD statement positions FATAR to the proper file; LABEL=(,NL) on the TAPEOUT DD statement requests an unlabeled scratch and causes FATAR to copy the TAPEIN data without labels. The NUMFILES=1 parameter restricts FATAR to copying one file. The input dataset name (from the TAPEIN DD statement) is used to name the output file.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=THIRDDS,UNIT=TAPE,VOL=SER=111111,
//           LABEL=3,DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,
//           LABEL=(,NL),DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        NUMFILES=1
```

**EXAMPLE 15:** When a tape is damaged (but readable) or suspect (prone to failure) you can copy the tape to another output volume with an “image copy” of the tape. This involves copying all data from the input to the output tape including all labels and the volume serial, creating an exact bit-for-bit copy. Since the output tape will have the same volume serial as the original, tape management systems and system catalogs will usually not have to be modified or updated. LABEL=(,BLP) is shown on the TAPEOUT DD statement in this example, but even if omitted the OUTBLP option is forced by LABELS=IMAGE so that all labels and data can be copied to the output tape. LABEL=(,BLP) is optional on TAPEIN.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=ANYDSN,UNIT=3490,VOL=SER=111111,
//           LABEL=(,BLP),DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=3490,
//           LABEL=(,BLP),DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        LABELS=IMAGE
```

**EXAMPLE 16:** Standard labeled tapes 333333, 444444, and 555555 comprise a multi-volume, multi-file tape set on 3480 cartridges, containing 17 files, and are to be copied to a 3590 Magstar cartridge. The VERIFY parameter causes the resulting 3590 volume to be analyzed for readability. If the FATAR tape management interface is enabled, it is used to get the full dataset name of every input file; the RENAME statement with NEWI=.ABC will cause the second level index of every input dataset name to be replaced with “ABC” when the files are opened on the output tape and CAT=ONLY causes those output files to be cataloged if they were cataloged to the input tape. For example, if an input dataset name was “PAYROLL.PROD.MASTER”, the output dataset becomes “PAYROLL.ABC.MASTER”.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPESUMM   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=3480,
//           VOL=SER=(333333,444444,555555),
//           DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=3590-1,
//           DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        VERIFY,CAT=ONLY
RENAME      LF=ALL,NEWI=.ABC
```

## 26.2 CONTINUED . . .

**EXAMPLE 17:  
COPY AND  
MODIFY A FILE**

A mailing list file on tape has a company name misspelled in many places; we want to copy the file and change the name wherever it appears. The REPLACE statement causes all occurrences of "INVENTION DATA PROCESSING" to be changed to "INNOVATION DATA PROCESSING" whenever it appears in column 17 of any record. Note the way that the character strings are continued; multiple C' or X' strings following a V=, S= or D= will simply be concatenated (up to 256 bytes). The MSGLEVEL=0 parameter suppresses the printing of the data portion of the modified blocks. CAT=RECAT will cause the file to be recataloged to the output tape.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=COMPANY.MAIL.MASTER,
//           DISP=(OLD,KEEP)
//TAPEOUT    DD     DSN=COMPANY.MAIL.MASTER,UNIT=CART,
//           DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY         MSGLEVEL=0,CAT=RECAT
REPLACE LF=1,B=ALL,P=17,V=C'INVENTION DATA',
        C'PROCESSING',D=C'INNOVATION DA',
        C'TA PROC',C'ESSING'
```

**EXAMPLE 18:  
COPY AND  
MODIFY A FILE**

Tape master file "MASTER" on labeled tape 111111 causes a run-time data exception when it is read by its processing program. A previous FATAR run has revealed that record 3 of block 27 contains "00735B4A" in hexadecimal, which is an invalid packed decimal number. It should contain "0073534C". The REPLACE statement causes this substitution to be made. CAT=ONLY will recatalog the dataset to the output tape if the input tape file was cataloged.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=MASTER,UNIT=TAPE,VOL=SER=111111,
//           DISP=OLD
//TAPEOUT    DD     DSN=MASTER,UNIT=TAPE,
//           DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY         CAT=ONLY
REPLACE LF=1,B=27.3,S=X'00735B4A',
        D=X'0073534C'
```

**EXAMPLE 19:  
RECOVER  
FROM DATA  
CHECKS**

Unlabeled tape 222222 is known to have data checks in a number of blocks. Copy the tape to another unlabeled scratch, retrying all data checks 256 times in each direction to insure that all possible attempts have been made to recover the data. If any block still cannot be read, the KEEP parameter will cause it to be written to the output tape anyway, but the REPLACE DCK statement will cause the first four bytes of those records to be set to hexadecimal "FFFFFFFF" (high values) which will signal the application program to perform special validation. The DCB parameters are provided on the TAPEIN DD statement so that the input records can be deblocked.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=NLINPUT,UNIT=TAPE,VOL=SER=222222,
//           LABEL=(,NL),DISP=OLD,DCB=(RECFM=FB,LRECL=240)
//TAPEOUT    DD     DSN=NLOUTPUT,UNIT=TAPE,
//           LABEL=(,NL),DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY         RETRY=256,KEEP
REPLACE DCK,F=1,B=ALL,P=1,D=X'FFFFFFFF'
```

## 26.2 CONTINUED . . .

**EXAMPLE 20:  
RECOVER  
FROM DATA  
CHECKS ON  
MULTI-VOLUME  
TAPE**

Dataset "LARGEDS" occupies three tape volumes "AAAAAA", "BBBBBB", and "CCCCC". Volume "BBBBBB" has developed read data checks, so we want to create a valid copy, dropping the data check blocks. With FATAR, it is not necessary to read the volumes preceding the first data check (in this case "AAAAAA"). The JCL below will produce output volumes which can be used to REPLACE "BBBBBB" and "CCCCC". However, it is up to the user to update the system catalog to reflect the new volume serials. If you have a tape management system, you must also do whatever manual updating is necessary to cause its records to properly reflect the new volume relationships. Note that this tape management update will be unnecessary if you copy the entire tape set (as in Example 16) or if you copy only the problem tape (as in Example 21).

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=LARGEDS,UNIT=TAPE,
//           VOL=SER=(BBBBBB,CCCCC),DISP=OLD
//TAPEOUT    DD     DSN=LARGEDS,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY
```

**EXAMPLE 21:  
RECOVER  
FROM DATA  
CHECKS**

This is another way to recover from the data checks in example 20 above. It does not require any catalog or tape management updates since it creates an exact copy ("image copy") of the bad tape, including its volume serial; in other words, the output of this run is a tape with the same volume serial as the original tape, so it should be externally re-labeled and the original tape discarded. Since tape "BBBBBB" is the middle tape of a multi-volume set, we know that it is completely filled with data; the output tape must be as large or larger than the original so that FATAR can WRITE all of the data and labels before the end-of-tape is detected, or the copy will be invalid. FATAR will correct the trailer label block counts if blocks are dropped because of data checks.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=LARGEDS,UNIT=TAPE,
//           VOL=SER=BBBBBB,DISP=OLD
//TAPEOUT    DD     DSN=LARGEDS,UNIT=TAPE,DISP=(NEW,KEEP),
//           LABEL=(,BLP,EXPDT=98000)
//SYSIN      DD     *
COPY        LABELS=IMAGE
```

**EXAMPLE 22:  
RECOVER  
DATA FROM  
AN  
OVERWRITTEN  
CARTRIDGE**

A cartridge tape with valid data on it was accidentally used for output, but the new file is smaller than the original file; it is desired to recover as much of the original data file as possible. LABELS=EOD causes FATAR to search for the first hardware EOD mark on the tape and recover data that follows it. Since header labels are not available for the overwritten file, the DCB characteristics of the original data should be provided on TAPEIN, if known. The TAPEOUT DD statement specifies a labeled scratch tape with the dataset name of the original dataset. The first data block is likely to cause a permanent data check if it is partially overwritten; if this is the only error it can be ignored. CAT=RECAT will cause the dataset to be recataloged to the output tape (be sure to specify the correct dataset name on the TAPEIN DD statement). If there are additional original files beyond the overlaid file, FATAR will automatically recovery them as well, using the dataset name and DCB information in their header labels to copy and catalog them to the output tape. See Section 24.2 for additional details on overwritten tapes. LABELS=EOD is not supported for round tapes and 3480 cartridges.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=ORIGINAL,UNIT=TAPE,
//           VOL=SER=BBBBBB,DISP=OLD,
//           DCB=(RECFM=FB,LRECL=125,BLKSIZE=32750)
//TAPEOUT    DD     DSN=ORIGINAL,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        LABELS=EOD,CAT=RECAT,BLP
```



## 26.2 CONTINUED . . .

**EXAMPLE 23:  
RECOVER  
DATA FROM  
AN  
OVERWRITTEN  
TAPE**

A round tape or 3480 cartridge with valid data on it was accidentally used for output, but the new file is smaller than the original file; it is desired to recover as much of the original data file as possible. The new file will occupy four physical files on the tape (header labels, data, trailer labels, and extra tape mark), so the LABEL=(5,BLP) will position FATAR immediately beyond the extra tape mark, at the beginning of the remaining original data. FATAR will process this as unlabeled data since it is not positioned to label records. Since labels are not available, the DCB characteristics of the original data should be provided on TAPEIN, if known. The TAPEOUT DD statement specifies a labeled scratch tape with the dataset name of the original dataset. The first data block is likely to cause a permanent data check if it is partially overwritten; if this is the only error it can be ignored. NF=1 causes FATAR to read only the original data file. CAT=RECAT will cause the dataset to be recataloged to the output tape (be sure to specify the correct dataset name on the TAPEIN DD statement). See Section 24.2 for additional details on overwritten tapes.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=ORIGINAL,UNIT=TAPE,
//              VOL=SER=BBBBBB,LABEL=(5,BLP),DISP=OLD,
//              DCB=(RECFM=FB,LRECL=125,BLKSIZE=32750)
//TAPEOUT    DD     DSN=ORIGINAL,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        NF=1,CAT=RECAT
```

**EXAMPLE 24:  
COPY FILES  
FROM AN  
OVERWRITTEN  
TAPE**

A round tape or 3480 cartridge with multiple files on it was accidentally used for output, but the new file is smaller than the original first file; it is desired to copy the original files beyond the first. The new file will occupy four physical files on the tape (header labels, data, trailer labels, and extra tape mark) and the original first file will occupy two more (data file and trailer labels) so the LABEL=(7,BLP) will position FATAR to the header labels of the original second file. FATAR will process this as a labeled tape; DCB information will be copied from the labels of the copied files. FATAR can copy dataset names from the original tape labels, but these datasets will no longer be recorded in your tape management system so FATAR will not be able to get the full original dataset names; if the original names are longer than 16 characters, the RENAME statements can be used to specify the correct names; CAT=RECAT will cause them to be recataloged to the new output tape. See Section 24.2 for additional details on overwritten tapes.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=ORIGINAL,UNIT=TAPE,
//              VOL=SER=BBBBBB,LABEL=(7,BLP),DISP=OLD
//TAPEOUT    DD     DSN=NOTUSED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        CAT=RECAT
RENAME      LF=3,NEWN=ORIGINAL3
RENAME      LF=4,NEWN=ORIGINAL4
```

**EXAMPLE 25:  
COPY AN  
IMPROPERLY  
CLOSED TAPE**

Due to a system crash or power failure, tape 333333 was not properly closed and has no trailing tape mark or labels; it cannot be read with standard access methods. A previous FATAR analyze job shows that block 742 is the last good block of the proper length; a data check occurs immediately after it. The job below will copy the tape up to and including block 742 onto a scratch tape with proper labels.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=A.NAME,UNIT=TAPE,
//              VOL=SER=333333,DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY        ENDAFTER=(LF=1,B=742)
```

## 26.2 CONTINUED . . .

**EXAMPLE 26:  
EXCLUDE FILES  
FROM A MULTI-  
FILE COPY**

Multi-file tape 444444 is to be copied to a scratch tape, but files 3 and 5 are not to be copied. The DROP statements with B=ALL will cause those files, including all of their labels and tape marks to be entirely omitted from the output tape. This means that file 4 from the input tape becomes file 3 on the output, file 6 from the input becomes file 4, file 7 comes file 5, etc. The VERIFY parameter causes the output tape to be read back to verify readability, number of files, labels, etc.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,
//           VOL=SER=444444,DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY         VERIFY
DROP LF=3,B=ALL
DROP LF=5,B=ALL
```

**EXAMPLE 27:  
DROP AN  
EXTRA TAPE  
MARK**

Because of a malfunctioning tape drive, labeled tape 555555 has an extra tape mark following the trailer labels for file 3. This makes all files past 3 unusable since data management expects an exact sequence of labels and tape marks; FATAR also cannot process the whole tape as labeled, but is capable of processing part of it. Two techniques are shown for recovering from this error:

The first technique causes an exact copy of the tape to be created, but without the extra tape mark. Since the tape mark looks like a file with no data blocks, the DROP statement shown will omit that file and therefore drop the tape mark (it is file 10 since the three standard label files preceding it appear to be three files each in LABELS=NO mode). The output tape will have the serial number of the original tape and should be externally re-labeled.

```
//FATAR      EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,
//           VOL=SER=555555,DISP=OLD
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,
//           LABEL=(,BLP),DISP=(NEW,KEEP)
//SYSIN      DD     *
COPY         LABELS=IMAGE
DROP LF=10,B=ALL
```

The second technique involves copying the data up to and beyond the extra tape mark in two separate FATAR steps, but processing in label mode. The first step copies the first three good files. LABEL=(11,BLP) in the second step positions the input tape to the header labels of the file beyond the tape mark, and LABEL=4 on the output tape causes FATAR to start the copy in the right place. The second step then continues to the end of the tape. The output tape will retain its original serial number, and labels will be properly updated if needed. If the FATAR tape management interface is enabled, it will be used to get the full dataset name of each input dataset; CAT=RECAT will cause them to be cataloged to the output tape.

```
//FATAR1     EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=555555,
//           LABEL=(1,BLP),DISP=(OLD,PASS)
//TAPEOUT    DD     DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,PASS)
//SYSIN      DD     *
COPY         NUMFILES=3,CAT=RECAT
//FATAR2     EXEC   PGM=FATAR,REGION=0M
//SYSPRINT   DD     SYSOUT=*
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=555555,
//           LABEL=(11,BLP),DISP=(OLD,KEEP)
//TAPEOUT    DD     DSN=NOT.USED,DISP=(NEW,KEEP),
//           VOL=REF=*.FATAR1.TAPEOUT,LABEL=4
//SYSIN      DD     *
COPY         CAT=RECAT
```

## 26.2 CONTINUED . . .

**EXAMPLE 28:  
FATAR AS A  
FATS SUB-TASK  
COPY TO  
MAGSTAR 3590**

FATAR may be run as a sub-task of FATS, and FATS parameters may be used to call for multiple tape volumes on the same drive. However, each tape is treated as a single volume, not as a multi-volume set. FATAR will be re-initialized for each tape processed. LABEL=(,BLP) is used on the TAPE1 DD statement since the tapes will probably have different dataset names, and "DDDDDD" is a dummy volume serial which will not be used. The VOL= parameter will cause tapes 111111, 222222, 333333, and 444444 to be copied. A console mount message will be issued for each tape requested. A 3590 scratch output tape will be mounted for each input tape, and all files copied to it. A full FATAR printout will be printed for each tape, and the FATS summary will show the results for all tapes. If the FATAR tape management interface is enabled, it will be used to get the full dataset name of each input dataset; CAT=RECAT will cause them to be cataloged to the output tape.

```
//FATFTR      EXEC   PGM=FATS
//SYSPRINT    DD     SYSOUT=*
//SYSPRIN1    DD     SYSOUT=*
//SUMMPRT     DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//TAPE1       DD     DSN=ANYDSN,UNIT=(TAPE,,DEFER),
//              VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP,EXPDT=98000)
//TAPE1OUT    DD     DSN=ANYDSN,UNIT=(3590-1,,DEFER),
//              DISP=(,KEEP)
//SYSIN       DD     *
              ANALYZE(1) VOL=(111111,222222,333333,444444),CAT=RECAT
```

**EXAMPLE 29:  
FATAR AS A  
FATS SUB-TASK  
MULTIPLE  
TAPES AND  
DRIVES**

When FATAR is run as a sub-task of FATS, FATS may invoke multiple copies of FATAR to perform operations on separate tape drives. Each FATAR execution will produce a separate report on its own SYSPRINx DD statement. Multiple volumes may be requested on each drive. The example below will request volumes ABC101, ABC102, and ABC103 on TAPE1, and volumes 000123 and 000125 on TAPE2 (copying the files on them to a scratch tape on TAPE2OUT).

```
//FATFTR      EXEC   PGM=FATS
//SYSPRINT    DD     SYSOUT=*
//SYSPRIN1    DD     SYSOUT=*
//SYSPRIN2    DD     SYSOUT=*
//SUMMPRT     DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//TAPE1       DD     DSN=ANYDSN1,UNIT=(TAPE,,DEFER),
//              VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP)
//TAPE2       DD     DSN=ANYDSN2,UNIT=(TAPE,,DEFER),
//              VOL=SER=EEEEEE,DISP=OLD,LABEL=(,BLP)
//TAPE2OUT    DD     DSN=NOT.USED,UNIT=TAPE,DISP=(,KEEP)
//SYSIN       DD     *
              ANALYZE(1) VOL=ABC101,VOLINCR=1,MAXVOLN=3
              ANALYZE(2) VOL=(000123,000125),VERIFY
```

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

**30.0 FATSCOPY FUNCTIONAL DESCRIPTION****WHAT IS  
FATSCOPY?**

FATSCOPY uses the copying facilities of FATAR to automate stacking of datasets on high-capacity tape volumes and migration to new types of tape. Only datasets with IBM standard labels (LABEL=SL) can be copied.

FATSCOPY can select the datasets to be copied or migrated from your system catalogs. You specify the datasets by:

- dataset name prefixes or masks and/or
- tape volume serial prefixes or ranges and/or
- catalog tape device type (3400, 3480, 3490E or 3590)

Once the datasets are selected from the catalog, if you have a supported tape management system, FATSCOPY will access information from your tape management system relating to each dataset. FATSCOPY currently supports CA-1 and DFSMSrmm. You can then additionally filter the datasets to be copied based on TMS information, including:

- creation date
- expiration date
- creating job name and/or step name
- dataset size

FATSCOPY can also be directed to copy entire tape volumes, by volser. The dataset names on the specified tape will be acquired from your tape management system (CA-1 or DFSMSrmm), and all files on that tape will be copied. If the tape volume specified is part of a multi-volume tape set, FATSCOPY will copy all volumes in that tape set and all datasets on the tape set, automatically.

FATSCOPY will dynamically allocate the input datasets it has selected and will then copy them to an output tape. Information recorded in your tape management system about each input dataset can optionally be propagated to the tape management records of the output datasets, and the input tapes can optionally be returned to scratch status for reuse.

**FATSCOPY  
TAPE  
STACKING**

Like all computer technology, tape technology keeps improving. Every few years a new generation of tape drives and tape media allows you to put more data in a tape cartridge with the same physical size. In the 1980s, 3480 tape cartridges held 200MB of data, but in the dawn of the 21st century cartridges of exactly the same physical size hold 80GB (80,000MB) or more. Undoubtedly, advances in technology will continue to expand the capacity of those tapes.

Unfortunately, many tape datasets are small or vary in size from one run to another, and most tape applications are designed to place only one dataset on a tape volume. The result is that many tapes in a tape library contain only a fraction of the data they are capable of holding. This results in inefficient use of the physical space in the library (a potentially expensive problem if an automated tape library (ATL/silo) is involved) and requires more cartridges, an additional expense.

These problems can be minimized and the resulting expenses avoided by putting more data on each cartridge. One way to do so is by placing multiple datasets on each cartridge, also called "stacking" datasets.

Although it is possible to automate dataset stacking by changing application JCL, it can be labor-intensive and error-prone to make the required changes and to monitor and correct them as application needs change. Also, it is subject to problems; for example, if one of the datasets to be stacked on tape is not created, perhaps due to an application ABEND, all the datasets to be stacked following that failed dataset will also fail.

## 30.0 CONTINUED . . .

**FATSCOPY  
TAPE  
STACKING  
(Continued)**

FATSCOPY avoids these problems and automates tape stacking by processing tape datasets that have already been created on tape. Your applications run without modification, creating datasets on tape as they always have. But periodically you can run FATSCOPY to collect all of those inefficient tape datasets and stack them as consecutive files on fewer tape volumes, releasing the inefficient input volumes as scratch tapes for reuse. The stacked datasets will be re-cataloged, so if they are needed as input they will be mounted and read transparently; the applications will not be aware that the stacking was done.

If input tapes were selected by volser instead of by cataloged dataset name, FATSCOPY will not stack tapes by default. All files on a given input tape or tape set will be copied to a single output tape (or tape set), but each new input tape will cause a new output tape to be used. However, you can request that these input files be stacked on the output tapes.

**FATSCOPY  
MIGRATION**

FATSCOPY can also be used when you are migrating from one type of tape device to another, such as when you install a new generation of faster, higher-capacity tape drives. Although much of your tape data will automatically migrate to the new tapes as new datasets are created on the new drives and old data expires, there will always be some long-term retention tape data which will be left on the old drives.

FATSCOPY can be used to copy this residual data to the new tapes, optionally stacking the datasets as well, allowing you to eventually remove the old technology drives.

## 31.0 FATSCOPY EXECUTION JCL

To execute FATSCOPY, the following JCL statements are required:

**EXEC  
STATEMENT**

The EXEC statement specifies the FATSCOPY program name and memory requirements (if your installation defaults are insufficient).

A region of 0M is recommended for FATSCOPY; the program will use no more memory than is required for a particular function.

Example:

```
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
```

**STEPLIB/JOBLIB  
DD STATEMENT**

A STEPLIB or JOBLIB DD statement will be required if FATSCOPY has been linked into a private library. It can be omitted if FATSCOPY is in a system library which can be accessed without a STEPLIB/JOBLIB statement (that is, a library in the system link list). This **must** be an APF authorized library.

**SYSPRINT DD  
STATEMENT**

SYSPRINT receives FATSCOPY messages, and is normally allocated to a SYSOUT dataset. Its DCB attributes are RECFM=FBA,LRECL=121. If BLKSIZE= is specified it must be a multiple of 121, otherwise it will default to 121 for SYSOUT or 1210 for other devices.

**SELRPT DD  
STATEMENT**

SELRPT will receive a report showing all datasets selected from the system catalogs (based on the SELECT/EXCLUDE statements present). It will be in catalog order, which is usually sorted by dataset name. It will also show tape management information for the dataset, and will indicate whether the dataset was actually selected for copying. It is usually a SYSOUT dataset and its DCB attributes are the same as for SYSPRINT.

**COPYRPT DD  
STATEMENT**

COPYRPT will receive a report showing all datasets actually copied to output tapes, in the order they were copied. It will display the output volser and file sequence number of each output file. It is usually a SYSOUT dataset and its DCB attributes are the same as for SYSPRINT.

**TAPESUMM DD  
STATEMENT**

TAPESUMM is optional, but if present receives a summary report of all input files in the order they were read, including dataset name, tape label information, block and byte counts, length of each file, minimum, maximum and average block sizes. It is usually a SYSOUT dataset and its DCB attributes are the same as for SYSPRINT.

**SYSUDUMP DD  
STATEMENT**

SYSUDUMP requests an abend dump if major errors occur (note that most internal abends in FATSCOPY are for the user's information only and do not cause dumps). SYSUDUMP is usually allocated to SYSOUT. Abend dumps are necessary for analysis of problems by Innovation. If you have the ABEND-AID product from COMPUWARE include the following so that a fully-formatted dump is produced:

```
//ABNLIGNR DD DUMMY
```

## 31.0 CONTINUED . . .

**TAPEOUT DD STATEMENT** The TAPEOUT DD statement specifies the output tape onto which FATSCOPY will copy the requested tape datasets. TAPEOUT will be ignored and should be omitted for a SIM (SIMULATE) operation. It must include:

DSN= a valid permanent dataset name, to meet MVS requirements. However, this name will not be used by FATSCOPY since actual output dataset names will be copied from the input tapes.

UNIT= a unit name which will allocate the desired type of output tape.

VOL= optionally, specify the volume serial of the output tape. If omitted, MVS will request a scratch tape. VOL= will usually be omitted.

DISP=(NEW,KEEP) do not specify CATLG since FATSCOPY will handle cataloging of the output datasets internally

RETPD= or EXPDT= specify one of these **only** if you intend to override the expiration date of **all** of the copied datasets.

If you specify STORCLAS= on the TAPEOUT DD, or you specify a DSN= which is assigned a SMS storage class by your SMS ACS routines, the output tape may be SMS-managed. However, FATSCOPY will not copy SMS class information from the input datasets to the output datasets, so the SMS classes specified or assigned will be used for all output datasets.

For example,

```
//TAPEOUT DD DSN=ANYNAME,UNIT=CART,DISP=(NEW,KEEP)
//TAPEOUT DD DSN=FILE.NAME,UNIT=3590-1,
//      VOL=SER=444444,DISP=(,KEEP)
```

**SYSIN DD STATEMENT** The SYSIN DD statement is the source of FATSCOPY control statements. It is normally a "DD \*" spool file, but can be any dataset with DCB characteristics RECFM=FB and LRECL=80.



**32.0 FATSCOPY CONTROL STATEMENTS****32.1 GENERAL****GENERAL  
RULES**

All FATSCOPY control statements must be input on 80-character records of which only columns 1 to 71 are used by FATSCOPY.

Every FATSCOPY control statement must begin with an operation keyword from the list below. This keyword may begin in column 1 or be preceded by any number of spaces. It must be followed by one or more spaces. Every operation keyword accepts a number of operands which follow those blanks and are separated by commas with no intervening spaces. Control statements may be continued onto multiple input records by following the last operand with a comma and a space, and starting the next operand on the next record in any column (preceded by spaces).

Any FATSCOPY control statement may contain comments; FATSCOPY stops its scan of the control statement whenever it finds an operand followed by a space (or comma-space for continuation), so the rest of the statement may be filled with any comment desired. Any control statement record containing an asterisk ("\*") in column 1 will be bypassed by FATSCOPY and may entirely contain comments.

FATSCOPY operation keywords:

COPY	Initiates the FATSCOPY operation and specifies general options
SIMULATE	Initiates a FATSCOPY simulation and specifies general options
SELECT	Specifies the datasets or tape volumes to be copied
EXCLUDE	Excludes certain datasets that would otherwise be selected by the SELECT statements.

## 32.2 COPY/SIM STATEMENT

COPY SIMULATE SIM	ABRINDEX=index *	,MAXFILE=nnnn
	,CAT=NO YES RECAT  <u>ONLY</u>	,MAXTOTFILE=nnnn *
	,EMCCCUNIT=esoteric	,METRIC  <u>NONMETRIC</u>
	,EXPDTGROUP=nnn  <u>Z</u>	,MULTIFILE=YES  <u>NO</u> *
	,EXPIRED=YES  <u>NO</u> *	,OFFSITE=YES  <u>NO</u>   <u>ONLY</u> *
	,HSMBAKMASK=hsmmask *	,OPCAP=nnnnM nnnnG
	,HSMMIGMASK=hsmmask *	,RETCODE
	,LASTFILE=nnnn  <u>1000</u> *	,TMSDATA= <u>COPY</u>   <u>NOCOPY</u>
	,LINECNT=nnnnn  <u>60</u>	,TMSIN=RMM CA1
	,MAJERR=nn  <u>1</u>	,TMSINPUT= <u>KEEP</u>   <u>SCRATCH</u>
	,MAXERR=nnnn  <u>10</u>	,TMSOUT=RMM CA1

\* these operands are ignored if SELECT ALLDSN statements are used to select tape volumes to copy.

A COPY or SIM Control Statement is required and must be the first statement in the input. However, all of its operands are optional. Operands may appear in any order on the statement. Note that many of the operands that are documented for the ANALYZE statement of FATAR (Section 23.2) are also supported on the FATSCOPY COPY statement, but they are rarely required so they are not documented here.

A SIMULATE (SIM) statement causes FATSCOPY to simulate the copy operation. It will scan the catalogs and tape management records according to your SELECT/EXCLUDE statements and it will display the datasets that it will select and show the order that they will be copied to output tapes. SIMULATE will not open TAPEOUT (which can be omitted) and it will not mount or read any input tapes.

## OPERANDS

**ABRINDEX=** Specifies the high-level index (1-8 characters) used to identify FDRABR Archive tape files to be automatically excluded. Such files are identified by matching against the dataset name as specified in Section 51 of the FDR manual.

Default: FDRABR, which is the default index used by FDRABR. If your installation uses a different ABR index, you should specify it here.

**Note:** FDRAPPL application backup datasets use the same naming convention as ABR Archive, except that the high-level index is usually an application prefix. If you are selecting datasets belonging to an application but want to exclude FDRAPPL backups for that application, specify ABRINDEX=applindex.

**CAT=** Controls cataloging of tape files copied to the output tape:

**NO** – output tape files will not be cataloged.

**YES** – output tape files will be cataloged if they are not already cataloged.

**RECAT** – output tapes files will be cataloged; if they are already cataloged, the catalog will be updated.

Default: CAT=RECAT

## 32.2 CONTINUED . . .

**EMCCCUNIT=** Specifies the esoteric name (1-8 characters) assigned to virtual tape drives managed by the EMC CopyCross product (a software virtual tape system), for example, EMCCCUNIT=VT90. Tape datasets cataloged to the device type associated with this esoteric name will be identified as VIRTUAL in reports and can be selected by the DEVTYPE=COPYCROS operand on a SELECT statement.

**EXPDTGROUP=** "nnn" specifies the allowable spread of expiration dates when FATSCOPY is stacking tape datasets. "n" may be 0-999; EXPDTGROUP=999 indicates that expiration date will be ignored when stacking datasets so that datasets with any expirations can be stacked together. The purpose of this is to group together datasets which will expire within a few days of one another, so that one dataset with an especially long retention will not cause the tape to be retained far past the expiration of other datasets on the tape. Datasets with special expirations (such as 99000 (catalog control) and 99nnn (cycle control)) will be grouped separately from datasets with explicit expirations unless EXPDTGROUP=999 is specified.

However, if datasets on a given input tape volume have varying expirations, FATSCOPY will group all of them on a single output tape based on the highest expiration of any of the selected datasets, to avoid mounting that input tape multiple times.

When SELECT ALLDSN statements are used to select input volumes by volser, expiration dates are not used to group tapes. By default, each input volume (or multi-volume tape set) will be copied to a separate output tape. But if EXPDTGROUP=999 is specified, then all input files on the selected input tapes will be copied to a single output tape or multi-volume tape set (except that the MAXFILE= operand may also be specified to limit the number of input files to be copied to a given output tape or tape set).

Default: EXPDTGROUP=7

**EXPIRED=** **YES** – if the input tapes are managed by the CA-1 tape management system, then both expired and non-expired datasets may be selected.

**NO** – only datasets which are not expired in the CA-1 tape management system may be selected.

Default: EXPIRED=NO

**HSMBAKMASK=**  
**HSMMIGMASK=** FATSCOPY will automatically exclude backup and migration tape datasets created by IBM's DFSMSHsm (HSM) by matching a mask against the names created by HSM. If your installation does not use the default IBM dataset names, you can specify a mask or a fully-qualified dataset name for the backup and migration datasets. See Section 32.4 for details on specifying a dataset name mask.

Default: FATSCOPY will exclude datasets matching these masks:

\*.COPY.HMIGTAPE.DATASET, \*.HMIGTAPE.DATASET,  
\*.COPY.BACKTAPE.DATASET, \*.DMP.\*.V\*.D+++++.T+++++,  
\*.BACKTAPE.DATASET and \*.DMP.T+++++.\*.D+++++.V\*

## 32.2 CONTINUED . . .

- LASTFILE=** If files are being stacked on an output tape (MAXFILE= is greater than 1), LASTFILE= controls how FATSCOPY operate when the end of the output tape is reached. "nnnnn" specifies a number of megabytes (MB). FATSCOPY will continue to add additional files to the current output tape until either MAXFILE= files have been written to the tape or the end of tape (EOT) is reached. If EOT occurs, FATSCOPY will calculate how much data from the current input file (the last file on the output tape) have been written. If the calculated megabytes is less than LASTFILE=, FATSCOPY will delete that file from the output tape, call for a fresh scratch output tape, and will recopy the current input file to the output tape. If more than LASTFILE= megabytes have been copied from the current input file to the output tape, FATSCOPY will call for a fresh scratch tape and continue to copy that input tape, creating a multi-file output tape.
- LASTFILE= does not apply to the first output file on a given output tape volume; if the current input file completely fills the output tape, it will always create a multi-volume output file.
- Default: LASTFILE=1000 which is 1000 Megabytes or 1 Gigabyte (GB).
- LINECNT=** "nnnnn" (1-32767) specifies the number of lines per page to print on all FATAR print datasets.
- Default: LINECNT=60
- MAJERR=** "nn" (1-99) specifies the maximum number of major errors (non-data check errors) that are allowed to occur on input tapes before the FATSCOPY step will stop attempting to copy tape files. Non-data check tape errors are usually equipment check errors or "intervention required" (tape no longer ready) conditions; these may indicate problems with the input tape drive or with the current input tape. Any major error on an input tape file will cause FATSCOPY to stop copying that file, but if the MAJERR= limit has not yet been reached it will attempt to copy another selected tape file (if any remain).
- Default: MAJERR=1, stop copying after the first major error.
- MAXERR=** "nnnn" (1-9999) specifies the maximum number of data checks that are allowed to occur on input tapes before the FATSCOPY step will stop attempting to copy files. Data checks usually indicate an error on the current input tape, but can also indicate errors with the tape drive. Any data check on an input tape file will cause FATSCOPY to stop copying that file, but if the MAXERR= limit has not yet been reached it will attempt to copy another selected tape file (if any remain).
- Default: MAXERR=10
- MAXFILE=** "nnnn" (1-9999) specifies the maximum file number to create on the output tape. If MAXFILE= is greater than 1, then tape stacking (writing multiple files on an output tape) will be done. Once "nnnn" files have been written on an output tape it will be dismounted and a fresh scratch tape will be mounted to receive new copied files. If MAXFILE=1, then no stacking will occur and a fresh scratch tape will be mounted for each output file.
- Default: files will be stacked on the output tape until it reaches its capacity.
- MAXTOTFILE=** "nnnn" (1-9999) specifies the maximum number of input tape files to copy in this FATSCOPY step. Once "nnnn" files have been selected for copying, FATSCOPY will stop searching the catalogs for additional files.
- Default: MAXTOTFILE=1000 input files

## 32.2 CONTINUED . . .

- NONMETRIC  
METRIC** When NONMETRIC is specified (or defaulted) FATSCOPY will list tape lengths and error positions and lengths in American units (feet and inches). When METRIC is specified FATSCOPY will list tape lengths and error positions and lengths in metric units (meters and centimeters).
- Default: NONMETRIC
- MULTIFILE=** Controls whether multi-file input tapes will be copied. When you are using FATSCOPY for tape stacking, this can be used to avoid copying tapes which already have stacked files (more than one file on the tape). This is based on the information selected from the catalog by your SELECT statements, so if you select only the first file on a multi-file tape, FATSCOPY will consider it to be a single file tape and will copy that file even if MULTIFILE=NO is specified.
- YES** – tapes containing multiple files will be copied. If you are using FATSCOPY to migrate tape files to a new type of tape, you will probably want to specify MULTIFILE=YES.
- NO** – tapes containing multiple files will be excluded.
- Default: MULTIFILE=NO – only single file tapes will be copied.
- OFFSITE=** Controls whether tape volumes which are recorded in your tape management system as being off-site (vaulted) will be copied. Usually tapes which are off-site should not be copied because, obviously, they are off-site and cannot be mounted. If you have a special reason for copying tapes which are marked as off-site and are sure they are available to be mounted, you can request that they be copied.
- YES** – tapes marked as off-site will be selected for copying, as well as on-site tapes. Be sure that your SELECT statements will select only those off-site tapes which you know are actually available for mounting.
- NO** – off-site tapes will be automatically excluded.
- ONLY** – only off-site tapes will be selected.
- Default: OFFSITE=NO – do not copy off-site tapes.
- OPCAP=** (used with SIMulate only) specifies the approximate uncompressed capacity of the output tapes that will be used when the FATSCOPY is run for real. If specified, the SIMULATE output will contain the approximate number of output tapes that will be required with no compression and with 2:1 compression. The value can be specified in megabytes (nnnnM) or gigabytes (nnnnG). For example, use OPCAT=10G if standard single-density 3590 tapes will be used for output.
- Default: the number of output tapes will not be estimated
- RETCODE** RETCODE causes FATSCOPY to terminate with a return code (condition code) of 12 rather than a U0200 or U0888 abend when serious errors are found.

## 32.2 CONTINUED . . .

- TMSDATA=** Specifies whether tape management data from each input file is to be copied to the output file after the input file is successfully copied.
- COPY** will copy the tape management data from the input file to the output file. This data includes: creation date, creating job name and creating step name. The expiration date of the input file will also be copied unless EXPDT= or RETPD= is specified on the TAPEOUT DD statement.
- NOCOPY** not copy the tape management data. The tape management records for the output file will show today as the creation date and will show the FATSCOPY job/step as the creating job/step. The expiration date of the output file will be the tape management default unless EXPDT= or RETPD= was specified on the TAPEOUT DD statement.
- Default: TMSDATA=COPY
- TMSIN=** FATSCOPY will normally determine the tape management system used for the input tapes by testing which FATS tape management interface you have installed, but if you installed both the DFSMSrmm and CA-1 interfaces, you must specify which system contains the tapes you are copying:
- RMM** – the input tape management system is IBM's DFSMSrmm.
- CA1** – the input tape management system is Computer Associate's CA-1 (TMS).
- Default: if both interfaces are installed, DFSMSrmm is assumed to be the input tape management. Otherwise, the installed interface is used.
- TMSINPUT=** Specifies how the input tape files are to be handled by your tape management system after they are successfully copied.
- KEEP** will retain the input tapes (note: if the retention of the tape file is “catalog control” and the file is recataloged to the output tape (via CAT=RECAT), the tape may still be scratched)
- SCRATCH** will scratch each input tape file after it is copied. If all files on an input tape are scratched, your tape management system will return the tape to scratch status.
- Default: TMSINPUT=KEEP
- TMSOUT=** Specifies that the tape management system used to manage the output tapes is different from the one used to manage the input tapes (from which FATSCOPY acquired tape information).
- RMM** – the output tape management system is IBM's DFSMSrmm.
- CA1** – the output tape management system is Computer Associate's CA-1 (TMS).
- Default: the output tape management system is the same as the input tape management system.

## 32.3 SELECT/EXCLUDE STATEMENT

```

SELECT      CATDSN=dsname
S           ALLDSN

           ,VOL=(volser,...,volser)

EXCLUDE     ,DEVTYPE=(devtype,...,devtype)
X           ,FILESEQ=nnnn
           ,CRDATE=yyyy.ddd
           ,EXPDATE=yyyy.ddd
           ,CRJOB=(jobname,...,jobname)
           ,CRSTEP=(stepname,...,stepname)
           ,CATLIMITGDG=nnn
           ,DSSIZE=nnnn

```

**SELECT  
STATEMENT**

The FATSCOPY input must include one or more SELECT statements to select datasets from your system catalogs or tape volumes from your tape management data. Each SELECT statement **must** contain either a CATDSN= operand or a ALLDSN operand; you cannot use both CATDSN= and ALLDSN on SELECT statements in the same FATSCOPY input.

If CATDSN= is specified, it gives the names of the datasets to be selected from system catalogs, and it may also contain other operands to limit the selected datasets. Only datasets cataloged to a tape device type will be selected.

The CATDSN= operand, and the VOL=, DEVTYPE=, and/or FILESEQ= operands, if they are specified, will be used to select datasets from your system catalogs. Additional operands, if specified, will be used to further filter the datasets selected from the catalog using information about those datasets from your tape management system. Note that if you specify the tape management-related operands, the datasets will be selected only if:

- you have a tape management system
- you have enabled the FATS tape management interface for your tape management system
- the dataset selected from the catalog is found in the tape management records for the tape volume as recorded in the system catalog.

A dataset must meet **all** of the selection operands specified on the SELECT statement in order for it to be selected.

If ALLDSN is specified, it must be accompanied by a VOL= operand, and no other operands. VOL= specifies a single tape volume serial which will be queried against your tape management database. All tape files on that input volume will be copied. If the tape volser is part of a multi-volume tape set, all datasets on all volumes in that set will be copied.

**EXCLUDE  
STATEMENT**

You may optionally include one or more EXCLUDE statements to exclude certain datasets that would otherwise be selected by a SELECT statement. **EXCLUDE statements must precede the SELECT statements to which they apply.** The EXCLUDE statement **must** contain a CATDSN= operand and may optionally contain other operands to limit the exclusion.

EXCLUDE statements cannot be used if SELECT ALLDSN statements are present.

## 32.3 CONTINUED . . .

**COMPARISON  
OPERATOR**

Although the syntax listings above show that all operands are followed by an equal sign (=), actually certain operands may be followed by comparison operator from the list below. Each operator has two forms, one or two special characters (such as = or >=) or an alphabetic equivalent surrounded by periods (such as .EQ. or .GE.). The operand descriptions which follow indicate which operands support which operators.

=	or	.EQ.	equal
<	or	.LT.	less than
>	or	.GT.	greater than
<=	or	.LE.	less than or equal to
>=	or	.GE.	greater than or equal to

**EXAMPLES:**

```
SELECT CATDSN=ABC.**
```

will select all cataloged tape datasets starting with "ABC."

```
SELECT CATDSN=XYZ.**.MASTER,VOL=B01*
```

will select all tape datasets starting with "XYZ." and ending in ".MASTER" which are cataloged to tape volumes starting with B01

```
EXCLUDE CATDSN=PAYROLL.HISTORY.**
SELECT CATDSN=PAYROLL.**
```

will select all cataloged tape datasets starting with "PAYROLL." except those starting with "PAYROLL.HISTORY."

```
SELECT CATDSN=(ABC.** , XYZ.**), CRDATE.GE.1998.001, CRDATE.LE.1999.365
```

will select all cataloged datasets starting with ABR. or XYZ. whose creation date is in 1998 or 1999.

```
SELECT CATDSN=XYZ.** , DSSIZE>25, DSSIZE<=75
```

will select all cataloged datasets starting with XYZ. whose size is from 26 to 75 megabytes.

**OPERANDS****CATDSN=  
ALLDSN**

CATDSN= specifies a fully-qualified dataset name or a filter to be used for generic tape dataset selection from system catalogs, as described in detail in Section 32.4. It is a required operand. CATDSN supports only the equal operator (= or .EQ.). When specified on a SELECT statement, it causes system catalogs to be searched for datasets which match. When specified on an EXCLUDE statement, it specifies datasets which will be excluded from the catalog search on SELECT statements which follow it.

If a fully-qualified name is specified, that name will be located in the system catalogs. Specification of a relative generation number for GDG datasets is supported (e.g., CATDSN=A.B(-1)).

If a filter is specified on a SELECT statement, then catalogs will be scanned for all cataloged datasets matching the filter (unless they are excluded by a preceding EXCLUDE statement).

CATDSN=\*\* will select all cataloged tape datasets. This should be used only when other operands are also specified to limit the datasets selected.

CATDSN=\*\* or any filter starting with a non-specific selection character will cause **all** catalogs on your system to be scanned, which may take a considerable amount of time.

**Additional considerations for CATDSN=filter are explained in Section 32.4.**

ALLDSN must be followed by VOL= which specifies a single tape volume serial. This tape volume will be looked up in your tape management database and all datasets on that tape will be copied. No operands other than VOL= are permitted with ALLDSN.



## 32.3 CONTINUED . . .

**VOL=** Specifies a tape volume serial or a volume serial prefix.

If used with CATDSN=, then only datasets cataloged to the volume(s) specified will be selected. The VOL operand supports all 5 comparison operators shown earlier in this section. You can specify:

V V V V V V a single tape volume serial, 1-6 alphanumeric characters

V V V V V \* a volume serial prefix (1-5 alphanumeric characters) followed by an asterisk

You can also specify a range of volume serials by specifying the VOL operand twice on the same SELECT or EXCLUDE statement, for example,

VOL>=400000,VOL<=444999

A list of volume serials or prefixes can be specified, enclosed in parenthesis and separated by commas; the dataset will be selected if it matches any one of the values specified. In this case, only the equal operand (= or .EQ.) is supported.

If VOL is omitted, then the volume serial will not be considered when processing the SELECT or EXCLUDE statement.

If ALLDSN is specified, then VOL=vvvvvv is **required** and must specify a single tape volume serial. This serial will be looked up in your tape management database and all datasets on that tape will be copied. If it is part of a multi-volume tape set, all tapes in the set will be copied; you can specify any volume in the tape set, and you must specify only one volume in the set (usually the first volume). If you have multiple SELECTs with volumes from the same set, the set will be copied multiple times.

**DEVTYPE=** Specifies a tape device type. Only datasets cataloged to the device type specified will be selected. DEVTYPE= must specify a valid MVS generic tape device type, such as 3400-6, 3480, 3490, or 3590-1. You cannot use esoteric names (such as TAPE or CART) since these names are not recorded in the system catalog. If multiple device types are specified, enclose them in parenthesis, separated by commas; the dataset will be selected if it matches any one of the values specified. DEVTYPE supports only the equal operator (= or .EQ.).

DEVTYPE=COPYCROS can be specified if the EMCCCUNIT= operand was specified on the COPY statement. Only datasets cataloged to the EMC CopyCross device type will be selected.

**FILESEQ=** Specifies a file sequence number. This is the file number of the file on a multi-file tape, originally specified by the LABEL=nnnn JCL parameter when the file was created. Only datasets cataloged as being on file number "nnnn" (1-9999) will be selected. The FILESEQ operand supports all 5 comparison operators shown earlier in this section.

You can also specify a range of file sequence numbers by specifying the FILESEQ operand twice on the same SELECT or EXCLUDE statement, for example,

FILESEQ>1,FILESEQ<123

**CRDATE=** Specifies a creation date, as a Julian date in the format yyyy.ddd. Only tapes created on the date(s) specified will be selected. CRDATE will be honored only if you have a tape management system, the FATS tape management interface is enabled, and the tape dataset selected from the catalog is found in your tape management records. The CRDATE operand supports all 5 comparison operators shown earlier in this section.

You can also specify a range of creation dates by specifying the CRDATE operand twice on the same SELECT or EXCLUDE statement, for example,

CRDATE>=1999.001,CRDATE<=1999.159

## 32.3 CONTINUED . . .

**EXPDATE=** Specifies an expiration date, as a Julian date in the format yyyy.ddd. Only tapes whose current expiration date matches this value, as recorded in your tape management system, will be selected. EXPDATE will be honored only if you have a tape management system, the FATS tape management interface is enabled, and the tape dataset selected from the catalog is found in your tape management records. The EXPDATE operand supports all 5 comparison operators shown earlier in this section.

You can also specify a range of creation dates by specifying the EXPDATE operand twice on the same SELECT or EXCLUDE statement, for example,

EXPDATE>=2001.001,EXPDATE<=2001.159

**CRJOB=** Specifies a creating job name (1 to 8 characters) or job name prefix. Only tapes which were created by this job, as recorded in your tape management system, will be selected. CRJOB will be honored only if you have a tape management system, the FATS tape management interface is enabled, and the tape dataset selected from the catalog is found in your tape management records. If multiple job names are specified, enclose them in parenthesis, separated by commas; the dataset will be selected if it matches any one of the values specified. The CRJOB operand supports only the equal operator (= or .EQ.). You can specify:

j j j j j j j a single job name, 1-8 alphanumeric characters

j j j j j j j \*a job name prefix (1-7 alphanumeric characters) followed by an asterisk

**CRSTEP=** Specifies a creating step name (1 to 8 characters) or step name prefix. Only tapes which were created by this step, as recorded in your tape management system, will be selected. CRSTEP will be honored only if you have a tape management system, the FATS tape management interface is enabled, and the tape dataset selected from the catalog is found in your tape management records. If multiple step names are specified, enclose them in parenthesis, separated by commas; the dataset will be selected if it matches any one of the values specified. The CRSTEP operand supports only the equal operator (= or .EQ.). You can specify:

s s s s s s s a single step name, 1-8 alphanumeric characters

s s s s s s s \*a step name prefix (1-7 alphanumeric characters) followed by an asterisk

**CATLIMITGDG=** "nnn" is used to limit the number of generations that CATDSN= will select from a GDG (generation data group), as described in Section 32.4.

**DSSIZE=** Specifies a dataset size, in megabytes. Only tape datasets which have this size, estimated from the blocksize and block count recorded in your tape management system, will be selected.

**32.4 DATASET NAME FILTERS**

The CATDSN= operand allows you to specify a filter which will be applied to tape datasets selected from system catalogs. The filter allows you a great deal of flexibility in specifying the names to be selected.

**FILTER  
CHARACTERS**

The dataset name filter is a string of characters which specify the datasets to be selected. Each character in the filter may represent a single character in the name of the datasets or may represent a variable number of characters. The filter characters may be:

- any alphanumeric character (A-Z or 0-9) or national character (\$ # @ in the US) represents itself and must appear in the indicated position in the dataset name
- / (slash) or % (percent) represents ANY single valid alphanumeric or national character
- | (vertical bar) represents any single alphabetic (A-Z) character
- + (plus) represents any single numeric (0-9) character
- ? (question) represents any single national (\$ # @ in the US) character
- . (period) is used to separate index levels.
- \* (asterisk) by itself as an index level indicates that the index level must exist in the selected name, but that it can contain any valid characters and can be any valid length (1 to 8 characters). However, if a single asterisk is combined with other characters in an index level, then it represents a variable length string (zero or more characters) at that point in the index level
- \*\* (double asterisk with no trailing period) represents zero or more characters or index levels. The filter routine will try to apply the remainder of the filter to the dataset name beginning with the current character and stepping through the name until it matches or until the end of the name is reached
- \*\*. (double asterisk with a trailing period) also represents zero or more characters or index levels, but the filter routine will try to apply the remainder of the filter to the dataset name at the beginning of each index level (if not currently at the beginning of a level, it will start at the next level).

These special cases apply to \*\* with periods:

- \*\*. at the beginning of the filter represents zero or more whole index levels at the beginning of the name
- \*\* at the end of the filter represents zero or more whole index levels at the end of the name.

The examples below are designed to illustrate the power and flexibility you have in using generic dataset name selection.

**EXAMPLE 1**

ACCOUNTS.PAY\*\*  
will select dataset beginning with "ACCOUNTS.PAY" such as

ACCOUNTS.PAY.OVERDUE  
ACCOUNTS.PAYROLL.CHECKS

**EXAMPLE 2**

USER1.\*.CNTL  
will select any 3-level dataset name with USER1 as the first level and CNTL as the third, such as

USER1.JCL.CNTL  
USER1.TEST.CNTL

**EXAMPLE 3**

USER+.\*\*.LIST  
will select any dataset whose first index is USERn (n is numeric) and whose last index level ends in LIST (including LIST by itself) with any number of index levels between them, such as

USER1.LIST  
USER3.ISPF.OUTLIST  
USER9.TEST.PRINT.MYLIST

**32.4 CONTINUED . . .****EXAMPLE 4    \*\*TEST\*\***

will select any dataset with the string TEST anywhere in the name, such as

```
ABC.TEST.GHI.XYZ
TEST2.LIST
REPORT.CURRENT.TEST
USER3.MYTEST
```

**EXAMPLE 5    \*\*A\*\*?\*B\*\***

will select any dataset whose name contains an A, a national character, and a B, in that order, such as

```
SYS1.A.X#B
APPLIC1.A@B.LOAD
APPLIC2.MASTER.BALANCE$.BOUNCE
```

**EXAMPLE 6    PAYROLL.%%11+\*.FILE**

will select any three-level dataset whose first index is PAYROLL, whose second index consists of any 2 characters, followed by any 2 alphabetic characters, followed by any 1 numeric character, and up to 3 more characters, and whose third index is FILE, such as

```
PAYROLL.DCLX5.FILE
PAYROLL.A4TV3LM.FILE
```

**EXAMPLE 7    PAYROLL.//\*\*\*.FILE.\*\***

will select any dataset whose first index is PAYROLL, whose second index consists of at least 2 characters, and which has any number of following index levels, one of which must be FILE, such as

```
PAYROLL.DCLX5.FILE
PAYROLL.AB.FACTORY1.FILE.OCT90
PAYROLL.FACTORY2.HISTORY.FILE
```

**EXAMPLE 8    \*\*.G++++V00**

will select all generations of any GDG (Generation Data Group).

**EXAMPLE 9    \*\*MASTER(-1)**

will select the —1 generation of any GDG whose name ends in MASTER, such as

```
PROD1.WEEKLY.WIDGET.MASTER(-1)
FINANCE.GLEDGER.XMASTER(-1)
```

This will only work when selecting datasets from the catalog (such as the CATDSN= operand).

**EXAMPLE 10    \*\***

will select all datasets.

## 32.4 CONTINUED . . .

**SELECTING  
DATASETS  
FROM THE  
CATALOG**

When generic dataset selection is used to select datasets from system catalogs, one or more system catalogs will be searched depending on the filter specified.

By default, the search will start with the master catalog; if aliases are encountered which match the filter, the associated user catalogs will be searched as well. If there are enough explicit characters at the beginning of the filter, the search may be limited to one or a small number of user catalogs. For example, a filter of AB\*\* means that only those user catalogs associated with aliases beginning with AB need be searched. However, if the filter begins with generic characters (such as \*\*AB), then every user catalog with an associated alias in the master catalog must be searched. Although the catalog search is designed to be as efficient as possible, reading the catalogs directly whenever possible, this may still be a time-consuming process.

GDGs (generation data groups) are treated specially when selected from a catalog:

- 1) the filter will be compared to the GDG base name (without the GnnnnVnn absolute generation). If it matches the base, then all generations in that GDG will be selected (unless limited by CATLIMITGDG= as shown below).
- 2) the filter will also be compared to the full name of each generation (including the GnnnnVnn). All generations which match the filter will be selected.
- 3) if the filter ends in a relative generation number in parenthesis (e.g., "PAYROLL.\*\*(-n)" or "PAYROLL.\*\* (0)") then only that generation will be selected from any GDGs selected (and the filter will not select any non-GDGs).
- 4) you can specify CATLIMITGDG=-n and only the (-n) generation of any GDGs which match the filter will be selected. However, the filter may also select non-GDG datasets.
- 5) you can also specify CATLIMITGDG=n and only the most recently created "n" generations of any GDGs which match the filter will be selected. The filter may also select non-GDG datasets.
- 6) For SMS-managed GDGs, any generations in rolled-off or deferred roll-in status will NOT be considered part of the GDG for selection; they will be treated as normal non-GDG datasets.

It is possible to assign aliases to non-VSAM datasets. These aliases are complete alternate names by which a dataset can be referenced and are not related to the aliases assigned to catalogs to control which datasets will be in a given catalog. The filter will never select alias names; it is tested only against the true name of every dataset.

STEPCAT and JOBCAT, if present, will normally be ignored. However, if the mask begins with a complete unqualified index, plus at least 1 more character after the period (e.g., CATDSN=PROD.X\*), and the STEPCAT/JOBCAT contains datasets which match the high-level index, then only the STEPCAT or JOBCAT will be searched for that request.

If an alias is encountered in a master catalog which points to a user catalog on a volume which is not currently mounted (or does not exist) that catalog will be bypassed.

**32.5 FATSCOPY SPECIAL CONSIDERATIONS****STACKING  
LIMITATIONS**

There is one important limitation that you must be aware of when stacking files on tape with FATSCOPY:

**The operating system will not allow two files on the same physical tape to be OPEN at the same time.**

If this should occur, the first OPEN will be successful but any subsequent OPEN for another file on the same tape will ABEND.

If an application program OPENS two tape files concurrently, and you use FATSCOPY to place both of those files on the same tape, the program will be unable to execute.

If two separate programs require tape files which you have placed on the same tape with FATSCOPY, and they happen to execute at the same time, one will work but the other will fail and must be rerun.

Unfortunately, there is no simple way to determine which tape files may have this problem, so FATSCOPY is not able to automatically avoid placing such files on the same tape.

**Therefore, it is the responsibility of the FATSCOPY user, through appropriate SELECT and EXCLUDE statements, to insure that tape files which may be needed by a single program or by programs executing concurrently are not placed on the same tapes.**

**FDR TAPES**

You can use FATSCOPY to copy backup tapes created by FDR, Innovation's backup software.

However, you should not use FATSCOPY to copy tapes created by FDRABR Archive or FDRAPPL Application Backup, because these tapes are recorded in an Archive Control File or Application Control File maintained by FDRABR, and these control files will not be updated when the tapes are copied by FATSCOPY. FDRABR includes tape utilities, FDRTCOPY and FDRTSEL, which can be used to copy these tapes and update the control files. See the FDR manual for details on the naming conventions used for these tapes and instructions for running FDRTCOPY and FDRTSEL. FATSCOPY will automatically exclude FDRABR Archive tapes; see the ABRINDEX= operand for details. However, FDRAPPL datasets usually begin with the high-level index of the application, so you may need to provide EXCLUDE statements for those datasets.

It is possible to use FATSCOPY to copy FDRABR Volume Backup tapes (full and incremental volume backups) because these tapes are simply cataloged in the ABR catalog, a standard MVS system catalog. When FATSCOPY copies these tapes and updates the ABR catalog, the copied tapes can be used by FDRABR. See the FDR manual for the naming conventions used for FDRABR Volume Backups. FATSCOPY *will not* automatically exclude FDRABR Volume Backup tapes.

**HSM TAPES**

It is not recommended that FATSCOPY be used to copy tapes created by HSM (DFSMSHsm) because they are recorded in data bases maintained by HSM which will not be updated when those tapes are copied. Consult IBM DFSMSHsm documentation for the format of HSM dataset names. If you select files with CATDSN=, FATSCOPY will automatically exclude HSM backup and migration datasets based on the default names created by IBM; see the HSMMIGMASK= and HSMBAKMASK= operands for details.

## 32.5 CONTINUED . . .

**CA-1  
CONSIDERA-  
TIONS**

If your tape management system is Computer Associates' CA-1 (also called TMS), the FATSCOPY support for selecting based on tape management criteria, scratching copied tapes, and updating tape management information is supported if you must install the FATS CA-1 interface as described in Section 90.3.

If TMSDATA=COPY is specified or defaulted, these fields will be copied from the CA-1 records of the input tape file to the output file:

- creation date and time
- creating job name, step name, and program name
- last used date and time
- last using job name and program name
- expiration date unless EXPDT= or RETPD= is specified on the TAPEOUT DD statement

if TMSINPUT=SCRATCH is specified, then FATSCOPY will update the expiration date of every successfully copied tape file to today's date. The expiration date of the FIRST dataset on the input tape will be set to the highest expiration date of any dataset on the tape (or multi-volume tape set). If all datasets on the tape or tape set have been copied by FATSCOPY, then CA-1 will scratch the tape the next time that you run CA-1 TMC maintenance.

**RMM  
CONSIDERA-  
TIONS**

If your tape management system is IBM's DFSMSrmm (RMM), the FATSCOPY support for selecting based on tape management criteria, scratching copied tapes, and updating tape management information in output tapes requires the RMM API (application program interface) which was added to RMM by IBM APAR OW30369 in 1998. In addition, it may require fixes to the API and some new features that were added by later maintenance. Innovation suggests applying the latest maintenance available for your level of RMM. You must also install the FATS RMM interface as described in Section 90.3.

If TMSDATA=COPY is specified or defaulted, these fields will be copied from the RMM records of the input tape file to the output file:

- creation date and time
- creating job name and program name
- last read date and time
- last write date and time
- expiration date (of the associated tape volume) unless EXPDT= or RETPD= is specified on the TAPEOUT DD statement

The creation date and creation time of the first dataset copied will be assigned as the RMM volume-level "assigned date" and "assigned time" for all output tapes created by FATSCOPY.

RMM does not provide an interface to update the creating step name.

Since RMM does not allow FATSCOPY to change expiration dates or delete individual datasets on tape volumes, if TMSINPUT=SCRATCH is specified, FATSCOPY will scratch the input tape only if all files on an input tape are being copied *in this FATSCOPY execution* and are eligible to be scratched.

**SMS-MANAGED  
TAPES**

The output tape used by FATSCOPY may be SMS-managed. However, FATSCOPY does not currently copy any SMS class information from the input datasets to the output datasets. You can specify STORCLAS= on the TAPEOUT DD statement, or you can specify a dataset name on the TAPEOUT DD which is assigned to a SMS-managed tape by your SMS ACS routines, to cause a SMS-managed tape to be mounted for output. The SMS classes specified or assigned will be used for all datasets copied.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**



## 33.0 FATSCOPY EXAMPLES

**MIGRATE  
CERTAIN  
DATASETS TO  
AN IBM 3590**

Cataloged tape datasets matching the dataset name mask will be copied and stacked on an IBM 3590 cartridge. The copied files will be recataloged to the output tape, and the input tapes will be returned to scratch status. These datasets will be stacked on a single output cartridge regardless of their expiration date, unless more than one cartridge is required to hold all the datasets. Since this is a migration, multi-file input tapes will be copied.

```
//STACK      EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT   DD  SYSOUT=*
//SELRPT     DD  SYSOUT=*
//COPYRPT    DD  SYSOUT=*
//TAPESUMM   DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//TAPEOUT    DD  DSN=OUTPUT,UNIT=3590-1,DISP=(,KEEP)
//SYSIN      DD  *
COPY        TMSINPUT=SCRATCH,EXPDTGROUP=999,MULTIFILE=YES
SELECT      CATDSN=TECH%.*.SAVE.**
```

**STACK FILES  
CATALOGED  
TO 3490E ON  
STORAGETEK  
9840**

Datasets created in 1999 and cataloged to the 3490E tape device type will be copied and stacked on StorageTek 9840 cartridges (esoteric name STK9840 has been defined in this installation to allocate a 9840 drive). If more than 500 files are selected, a fresh scratch tape will be mounted for each set of 500 output files. The copied files will be recataloged to the output tape, and the input tapes will be retained by your tape management system (you may wish to scratch them or delete them from tape management control later). Certain datasets (system datasets and ABR Volume backups) are excluded.

Datasets will be grouped by their expiration date: datasets that expire within a range of 7 days will be stacked together on one cartridge (so that all files on the cartridge will expire at about the same time). Datasets with special expirations, such as 99000 for catalog control, will be stacked on a separate cartridge.

```
//STACK      EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT   DD  SYSOUT=*
//SELRPT     DD  SYSOUT=*
//COPYRPT    DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//TAPEOUT    DD  DSN=OUTPUT,UNIT=STK9840,DISP=(,KEEP)
//SYSIN      DD  *
COPY        MAXFILE=500
EXCLUDE     CATDSN=SYS**
EXCLUDE     CATDSN=FDRABR.**
SELECT      CATDSN=**,DEVTYPE=3490,CRDATE>=1999.001,CRDATE<=1999.365
```

**SIMULATE  
STACKING OF  
DATASETS**

Dataset stacking will be simulated. FATSCOPY will select cataloged datasets matching the CATDSN= mask which were created by jobs beginning with PAY or INV and the listing will show all selected datasets. Since this may select datasets created by FDR Application Backup (FDRAPPL), an EXCLUDE statement for FDRAPPL datasets is provided.

```
//SIMULATE   EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT   DD  SYSOUT=*
//SELRPT     DD  SYSOUT=*
//COPYRPT    DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//SYSIN      DD  *
SIM
EXCLUDE     CATDSN=PROD*.V////////.B+++++/
SELECT      CATDSN=PROD**,CRJOB=(PAY*,INV*)
```

## 33.0 CONTINUED . . .

**COPY TAPE  
VOLUMES INTO  
A VTS**

All files on 3 selected input volumes will be copied into a VTS (virtual tape system) as specified by UNIT=VTS on the TAPEOUT statement. The 3 tape volumes will be identified from tape management information and all files on those tapes will be copied. Each input tape will be copied to a separate VTS virtual tape. If any of the tapes are part of a multi-volume tape set, then all files on all tapes in the set will be copied. Output files will be re-cataloged to the output tapes only if they were cataloged to the input tapes (CAT=ONLY is the default).

```
//VTS      EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD  SYSOUT=*
//SELRPT   DD  SYSOUT=*
//COPYRPT  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TAPEOUT  DD  DSN=OUTPUT,UNIT=VTS,DISP=(,KEEP)
//SYSIN    DD  *
COPY       CAT=ONLY
SELECT     ALLDSN,VOL=030427
SELECT     ALLDSN,VOL=010034
SELECT     ALLDSN,VOL=004961
```

**STACK TAPE  
VOLUMES  
ONTO AN IBM  
3590**

All files on 5 selected input volumes will be stacked onto a single IBM 3590 Magstar tape. The 5 tape volumes will be identified from tape management information and all files on those tapes will be copied and stacked on the output. If any of the tapes are part of a multi-volume tape set, then all files on all tapes in the set will be copied. All copied files will be cataloged to the output tape, regardless of the catalog status of the input files.

```
//STACK     EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD  SYSOUT=*
//SELRPT   DD  SYSOUT=*
//COPYRPT  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//TAPEOUT  DD  DSN=OUTPUT,UNIT=3590-1,DISP=(,KEEP)
//SYSIN    DD  *
COPY       EXPDTGROUP=999,CAT=RECAT
SELECT     ALLDSN,VOL=A03452
SELECT     ALLDSN,VOL=C19498
SELECT     ALLDSN,VOL=C19502
SELECT     ALLDSN,VOL=R32343
SELECT     ALLDSN,VOL=F00042
```

**40.0 FATS AND FATAR MESSAGES AND CODES****40.1 RETURN CODES**

- RETURN CODES** Both FATS and FATAR use the following completion or return codes:
- 0** Normal completion. No serious errors
  - 12** Serious errors have occurred and the RETCODE option has been specified (user abend U0200 or U0888 is issued otherwise).
  - 32** A trial version of the product is being executed and the trial period has expired (contact INNOVATION if you wish to continue your testing or install the production version).

**40.2 ABEND CODES**

- ABEND CODES** Both FATS and FATAR may issue the following user abend codes (most abends will be accompanied by an explanatory message):
- U0047** The FATS or FATAR program is not authorized. See message FATS112.
  - U0100** (FATAR only) TAPEIN DD statement missing, TAPEIN or TAPEOUT is not assigned to a tape unit, or an open failure on TAPEIN or TAPEOUT
  - U0200** (FATAR only) unrecoverable I/O error on TAPEOUT. If the RETCODE parameter is specified, a return code of 12 is issued instead of this abend.
  - U0401** Open error or I/O error on SYSIN
  - U0402** Open error or I/O error on a PRINT dataset
  - U0502** Error in FATS or FATAR control statements
  - U0888** Major errors on tape. For FATS, any condition causing abnormal completion of a FATS function (indicated by "ABNORM" in the FATS summary report). For FATAR, unrecoverable I/O errors on TAPEIN other than data checks, or MAXERR=/MAXTERR= values exceeded. If the RETCODE parameter is specified, a return code of 12 is issued instead of this abend.
  - U0913** Security for FATS (Section 11.8) or FATAR (Section 21.6) has been implemented and the user did not have authority to perform the requested function.

### 40.3 CONSOLE MESSAGES

**CONSOLE MESSAGES** The following write-to-operator (WTO) and write-to-operator-with-reply (WTOR) messages may be issued by FATS or FATAR. They will be written to route code 3 (tape pool console).

#### **FATSW01 REQUEST TO LABEL TAPE ON UNIT *uuuu* REPLY WITH SERIAL NUMBER**

**Reason:** The OPERATOR parameter has been specified or the SAVLAB function found that the tape was not a labeled tape

**Action:** Respond with a 1-to-6 character volume serial. FATS will left justify it, blank fill it, and label the tape on tape unit "uuuu".

#### **FATSW02 REPLY EOJ/KEOJ/*Kuuuu* TO TERMINATE**

**Reason:** FATS will always issue this WTOR at the start of every FATS run to allow the operator to control and terminate FATS

**Action:** It will not be necessary to reply to this message unless:

- 1) MULT was specified without MAXVOLN=
- 2) It is desired to terminate FATS before its normal termination, or
- 3) A particular tape must be halted. Valid responses are:

EOJ – FATS will terminate after all tapes which are currently in progress have completed. Any pending tape mounts must be satisfied.

KEOJ – FATS will halt all tapes in progress, cancel any pending mounts, and terminate with a U0888 abend or return code 12.

Kuuuu – Where "uuuu" is the 3 or 4-digit address of a tape drive allocated to FATS. Any mount or tape in progress on that drive will be halted. If appropriate, another tape may be requested on the drive. Other drives allocated to FATS will not be affected. If "uuuu" is not a drive in use by FATS, it will be ignored. The FATSW02 message will be reissued.

#### **FATSW03 FATS ANALYSIS OF TAPE ON UNIT=*uuuu* VOL=*vvvvvv* PERM ERRORS=*ppppp* TEMP ERRORS=*tttt***

**Reason:** The WTO parameter has been specified. This message will be issued at the end of processing for each tape, giving unit "uuuu", volume serial "vvvvvv", and the total number of permanent and temporary errors ("ppppp" and "tttt"). If a tape has major errors causing it to be halted prematurely, "ppppp" will be "MAJOR".

#### **FATSW04 MOUNT NEXT REEL ON UNIT=*uuuu* VOL=*vvvvvv***

**Reason:** Issued by FATS to request mounting of another tape on tape unit "uuuu". The volume serial "vvvvvv" may be an explicit serial or "SCRTCH" as appropriate.

**Action:** Mount the requested tape on unit "uuuu".

## 40.3 CONTINUED . . .

**FATSW05 FATS HAS STOPPED ON DATA CHECK ON UNIT=uuuu REPLY CONT, TERM, NOSTOP**

**Reason:** The STOP or STOPNUM= parameters have been specified and a permanent data check has occurred. FATS stops to allow visual inspection of the tape at the point of the error.

**Action:** Inspect the tape on tape unit "uuuu", if desired, re-ready the tape, and reply:

- CONT – To continue certifying/verifying, stopping again on the next data check
- TERM – To terminate processing of the tape
- NOSTOP – To continue certifying/verifying without stopping on further data checks

Note that if you remove the tape from the "READY" state to inspect it, you will probably be unable to re-ready it at the same spot and be forced to reply "TERM".

**FATSW06 PLEASE CLEAN UNIT=uuuu THEN (RE)MOUNT VOL=vvvvvv**

**Reason:** The MAXCLEAN= or ERRCLEAN= parameters were specified. If MAXCLEAN=n then "n" tapes have been processed on unit "uuuu" since the last cleaning. If ERRCLEAN=n then more than "n" errors have occurred on unit "uuuu" (the message will say REMOUNT).

**Action:** Clean tape unit "uuuu" and mount or remount the specified volume "vvvvvv"

**FATSW07 INVALID RESPONSE TO TERMINATE MESSAGE – RE-ENTER**

**Reason:** The response to message FATSW02 was not "EOJ", "KEOJ", or "Kuuuu". FATSW02 will be reissued.

**Action:** Reply with a valid response for message FATSW02

**FATSW08 VOL=vvvvvv MOUNTED,DSN=d...d,EXPDT=yyddd  
UNIT=uuuu REQUIRES VOL=xxxxxx-REPLY IGNORE, UNLOAD, TERM, SKIP  
UNIT=uuuu DSN NOT EXPIRED – REPLY IGNORE, UNLOAD, TERM, SKIP**

**Reason:** The VALIDATE= parameter was specified to validate input and/or output labeled tapes with LABEL=(,BLP). Either the wrong volume was mounted or the expiration date on an output volume was not reached. In the first line of the message, "uuuu" is the tape unit, "vvvvvv" is the actual volume mounted, "d...d" is the dataset name read from the tape, and "yyddd" is the expiration date. One of the next two formats of the message will be issued ("xxxxxx" is the volume expected).

**Action:** Reply with:

- IGNORE – To ignore the error and accept the volume mounted.
- UNLOAD – To unload the volume and request mounting of another tape
- TERM – To terminate FATS or FATAR
- SKIP – To terminate this volume and go on to the next (FATS only)

40.3 CONTINUED . . .

**FATSW11 FATAR FILE *ffff* OPERATION UNIT=*uuuu*,VOL=*vvvvvv*,  
BLOCKS=*bbbbbb*,PERM ERRS=*ppppp*,TEMP ERRS=*ttttt***

**Reason:** The WTO parameter has been specified. The message will be issued at the end of processing each input file, giving file number "*ffff*", unit "*uuuu*", volume serial "*vvvvvv*", input block count "*bbbbbb*", and the total number of permanent and temporary errors ("*ppppp*" and "*ttttt*") in that file. "Operation" will be "ANALYZED ON" (if TAPEOUT absent), "COPIED FROM" (if TAPEOUT present), or "VERIFIED ON" (if verifying TAPEOUT)

**Action:** None

**FATSW12 FATAR FILE *ffff* OPERATION UNIT=*uuuu*,VOL=*vvvvvv* REPLY CONT, TERM, OR NOSTOP**

**Reason:** The STOP parameter has been specified. The message will be issued at the end of processing each input file, giving file number "*ffff*", unit "*uuuu*", and volume serial "*vvvvvv*". "Operation" will be "ANALYZED ON" (if TAPEOUT is absent), or "COPIED FROM" (if TAPEOUT present).

**Action:** Reply with:

- CONT      – To continue with the next input file, stopping again at the end of that file
- TERM      – To terminate processing of TAPEIN
- NOSTOP   – To continue with the next file without stopping at the end of further files

**FATSW13 REPLY IS: TEXT**

**Reason:** Automatic reply to tape management message.

**FATSW14 INVALID MODIFY COMMAND GIVEN: *text*...**

**Reason:** "*text*" was specified in MODIFY (F) id,text operator command. "id" is the FATS jobname for which the MODIFY command was intended. "*text*" is not either "QUIT" or "Kuuuu" to terminate FATS when current tapes are done or to immediately terminate processing for tape unit "*uuuu*".

**Action:** None. Command is ignored.

**FATSW89 – *xxxx* — TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN *nnn* DAYS.  
– *xxxx* – TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRED -----  
PLEASE CONTACT INNOVATION.**

**Reason:** "*xxxx*" (FATS or FATAR) trial will expire in "*nnn*" days or "*xxxx*" trial has expired. The version of FATS or FATAR that you are running is a trial version and has expired or will expire shortly.

**Action:** If you have purchased FATS/FATAR and have received the tape, make sure that it was properly installed and the trial version has been deleted. Then rerun the job. If you have not yet purchased FATS/FATAR, call INNOVATION DATA PROCESSING.

**40.4 PRINT MESSAGES**

**PRINT MESSAGES** FATS and FATAR may issue the following messages to their datasets (SYSPRINT or SYSPRINx).

**FATS001 TAPEIN DD STATEMENT MISSING**

**Reason:** The DD statement for TAPEIN is missing from the job step JCL.

**Action:** FATAR is terminated with a U0100abend. Correct the JCL and resubmit the job.

**FATS002 TAPEIN DEVICE IS NOT A TAPE UNIT**

**Reason:** The DD statement for TAPEIN does not reference a supported tape unit type.

**Action:** FATAR is terminated with a U0100abend. Correct the JCL and resubmit the job.

**FATS003 TAPEIN COULD NOT BE OPENED**

**Reason:** The open of the input tape on TAPEIN failed.

**Action:** FATAR is terminated with a U0100abend. Check the printout for operating system messages indicating the cause.

**FATS005 TAPEOUT DEVICE IS NOT A TAPE UNIT**

**Reason:** The DD statement for TAPEOUT does not reference a supported tape unit type.

**Action:** FATAR is terminated with a U0100abend. Correct the JCL and resubmit the job.

**FATS006 TAPEOUT COULD NOT BE OPENED**

**Reason:** The open of the output tape on TAPEOUT failed.

**Action:** FATAR is terminated with a U0100abend. Check the printout for operating system messages indicating the cause.

## 40.4 CONTINUED . . .

**FATS010 CONTROL CARD ERROR — REASON X**

**Reason:** A syntax or usage error has been detected in the control statement printed immediately preceding the message. Only one FATS010 message will be printed for each input statement (multiple errors will not be diagnosed). "x" is a letter or number indicating the type of error detected:

- 1 – Open failed for the control statement dataset (SYSIN or SYSINx)
- 2 – Control statement is completely blank
- 3 – An invalid operation keyword was found on the control statement
- 4 – More than one ANALYZE/COPY control statement was found.
- 5 – The control statement required at least one keyword parameter but did not contain one (for all statements except ANALYZE/COPY).
- 6 – A specific keyword parameter was required on the control statement but was not found or a required parentheses was missing or a keyword was misspelled.
- 7 – An ANALYZE/COPY control statement was required but not found.
- 8 – T= on TABLE statement in error. Letter previously used.
- 9 – A numeric value for a keyword parameter exceeds the allowed maximum for that keyword
- A – A value for a keyword parameter was zero or non-numeric
- B – Continuation card required but not found
- C – B= keyword in error. Block range specified but not in ascending order
- D – D=, V=, or S= keyword in error. Length of data is more than 256 bytes or characters.
- E – D=, V=, or S= keyword in error. Data field must contain at least 1 byte or character.
- F – ENDAFTER= keyword error.
- G – Control statement table is too small to contain all information from the control statements present. Increase the table size via the EXEC JCL statement PARM field.
- H – Error in X ' or C' data: no closing apostrophe, invalid HEX digits, or odd number of HEX digits.
- J – Invalid data string. Must start with X ' or C'
- K – Error in T= keyword. On SCAN statement, must be A-Z and defined in a previous table statement (except for P and Z). On table statement, must be A-O or Q-Y and not previously used.
- L – EXITNAME was not found or length of EXITNAME is zero.
- M – Dataset name specified by NEWN= on a RENAME statement is invalid.
- N – Dataset name mask specified by NEWI= or CATDSN= on a SELECT, EXCLUDE or RENAME statement is invalid.
- O – Invalid mask on SELECT statement or in HSMxxxMASK= operands.
- P – Invalid volume list or conflicting selection parameters on SELECT statement
- Q – invalid value for ABRINDEX= operand.
- R – invalid value for OPCAP= operand.

**Action:** FATAR will abend with a U0502 abend code after all control statements have been scanned.

**FATS011 INVALID CONTINUATION CARD**

**Reason:** FATSCOPY detected an invalid continuation on an input statement.

**Action:** FATSCOPY will abend with a U0502 abend code. Correct the errors and re-execute.



**40.4 CONTINUED . . .****FATS013 ONE OR MORE ERRORS DETECTED – FATAR TERMINATED**

**Reason:** Message FATS010 was issued for one or more control statement errors.

**Action:** FATAR will abend with a U0502 abend code. Correct the errors and re-execute.

**FATS014 NO CONTROL CARD DATASET. DEFAULTS ASSUMED**

**Reason:** No SYSIN DD statement for control statement input was present in the job step JCL (if FATAR was invoked by FATS, no SYSINx DD was present).

**Action:** All defaults are assumed for all ANALYZE statement parameters. If FATAR was invoked by FATS, any FATAR parameters specified in FATS statements will override these defaults.

**FATS016 ANALYSIS TERMINATED DUE TO MAJOR ERROR**

**Reason:** An unrecoverable hardware error has occurred on the input tape drive or channel. Preceding messages will give details of the error.

**Action:** FATAR is terminated with a U0888 abend or return code of 12. Contact INNOVATION if you need assistance in determining the cause.

**FATS017 MAXIMUM ERRORS EXCEEDED – FATAR TERMINATED**

**Reason:** The number of unrecoverable data checks exceeded the MAXERR= parameter, or the number of recoverable data checks exceeded the MAXTERR= parameter.

**Action:** FATAR is terminated with a U0888 abend or return code of 12.

**FATS019 FATAR MUST BE IN AN AUTHORIZED LIBRARY. PROGRAM TERMINATED**

**Reason:** FATAR must be executed as an APF authorized program. Either the program library containing FATAR was not marked as an APF authorized library on your MVS system, or your STEPLIB or JOBLIB DD statement has nonauthorized libraries concatenated to the FATAR program library, or FATAR has been relinked as nonauthorized (AC=0).

**Action:** FATAR will terminate with a U0047 ABEND.

**FATS020 ANALYSIS TERMINATED AT TAPEMARK SEQUENCE**

**Reason:** If labels are being processed, two consecutive tape marks were read after a label file. If labels are not processed and NUMFILES=0 was specified or defaulted, two consecutive tape marks were read.

**Action:** Processing of the input tape is completed.

**FATS021 ANALYSIS TERMINATED BY EOVLABELS DETECTED IN LABELS=NO MODE**

**Reason:** If labels are not being processed and NUMFILES=0 was specified or defaulted, FATAR has detected what appears to be IBM or ANSI EOVLABELS.

**Action:** Processing of the input tape is completed.

**40.4 CONTINUED . . .****FATS022 ANALYSIS TERMINATED AFTER SPECIFIED NUMBER OF FILES**

**Reason:** NUMFILES=n was specified, and “n” logical files have been read.

**Action:** Processing of the input tape is completed.

**FATS023 ANALYSIS TERMINATED BY ENDAFTER OPTION**

**Reason:** The file and block specified by the ENDAFTER= parameter was read.

**Action:** Processing of the input tape is completed.

**FATS024 ANALYSIS TERMINATED BY LOGICAL END OF FILE**

**Reason:** FATAR attempted to switch to a new input tape volume (due to EOVLABELS read, or end-of-file in LABELS=OS mode), but no more volumes were given.

**Action:** Processing of the input tape is completed.

**FATS025 ANALYSIS TERMINATED BY OPERATOR REQUEST**

**Reason:** The operator replied “TERM” to the FATSW12 message or FATSW08 message.

**Action:** Processing of the input tape is halted.

**FATS026 TAPEIN DATA SUCCESSFULLY COPIED TO TAPEOUT VOLUME VVVVVV**

**Reason:** The TAPEOUT DD statement was present, and reading of TAPEIN data completed successfully. All input data not “DROPPED” has been written to TAPEOUT. The last or only volume of TAPEOUT was “vvvvvv”.

**Action:** None

**FATS027 UNLABELED TAPEOUT REQUIRED MORE THAN ONE VOLUME – VERIFY CANCELLED**

**Reason:** TAPEOUT is an unlabeled tape (LABEL=(,NL) or (,BLP)), the VERIFY parameter was specified, and TAPEOUT overflowed to more than one volume

**Action:** FATAR cannot verify multi-volume unlabeled tapes. The VERIFY will not be performed.

**FATS028 ANALYSIS TERMINATED DUE TO INVALID LABEL**

**Reason:** Labels were being processed on TAPEIN, but a valid IBM standard or ISO/ANSI or user label was not read when one was expected.

**Action:** The invalid data will be printed, and FATAR will be terminated with a U0888 abend or return code 12

## 40.4 CONTINUED . . .

**FATS030 EOT ON TAPEOUT BUT VOLUME SWITCH NOT ALLOWED – FATAR TERMINATED**

**Reason:** Labels are being processed on TAPEIN, the current file being copied contains user labels, and the SUPULAB parameter (suppress user labels) was not specified. User labels may be required with the EOVS (end-of-volume) and header labels written when the file crosses volumes, but FATAR has no way to generate user labels acceptable to the user application.

**Action:** FATAR is terminated with a U0888 abend or return code 12

**FATS033 PERMANENT I/O ERROR ON TAPEOUT – FATAR TERMINATED**

**Reason:** FATAR uses standard system error recovery on TAPEOUT. An uncorrectable I/O error occurred (messages will follow to detail the error which occurred).

**Action:** FATAR will be terminated with a U0200 abend or return code 12. Consult the FATAR messages and operating system messages for details of the error. Errors may be avoided by cleaning the TAPEOUT tape drive, and using an output tape certified by FATS or other means.

**FATS034 NOISE BLOCK HAS BEEN READ**

**Reason:** A data check block too small to be considered a valid data block has been read (will not occur on tape cartridge drives).

**Action:** No recovery attempts will be made. The block will be immediately flagged as a permanent data check.

**FATS035 TAPE DRIVE NOT CAPABLE OF REQUIRED DENSITY/MODE**

**Reason:** The tape drive has reported that it is not capable of reading the input tape. This may occur when a tape written at a certain density (such as 6250 BPI for round tapes) or format (such as 3490E format) is mounted on a drive that is not capable of reading that density or format. This error may also be caused by an uninitialized tape.

**Action:** FATAR is terminated with a U0888 abend. Determine the actual density of the tape, and change the TAPEIN DD statement to allocate an appropriate tape drive.

**FATS036 BLOCK GREATER THAN *nnnnnn* BYTES READ – FATAR TERMINATED**

**Reason:** A tape block longer than "*nnnnnn*" bytes (as specified by the BLKSIZE= parameter, DEFAULT 262144 or 256K) has been read. This form of the FATS036 message is produced only if an output tape is being written, i.e., a TAPEOUT DD statement is present.

**Action:** FATAR is terminated with a U0888 abend.

**FATS036 \*\* WARNING \*\* BLOCK GREATER THAN *nnnnnn* BYTES READ –  
BYTE COUNTS WILL BE INCORRECT**

**Reason:** A tape block longer than "*nnnnnn*" bytes (as specified by the BLKSIZE= parameter, default 262144 or 256K) has been read. This form of the FATS036 message is produced when no output tape (TAPEOUT DD statement) is present.

**Action:** The excess data is discarded, and FATAR continues with the next block. However, byte counts printed will not include the excess bytes.

## 40.4 CONTINUED . . .

**FATS037 TAPE LABELS IN ISO/ANSI FORMAT – TRANSLATED TO EBCDIC**

**Reason:** FATAR has detected that the labels on TAPEIN are in ISO/ANSI format (“ANSI labels”).

**Action:** The labels will be translated to EBCDIC for printing or processing. Also, the associated data file is assumed to be in ISCII/ASCII and will be translated to EBCDIC.

**FATS038 DATA FILE IN ISCII/ASCII – TRANSLATED TO EBCDIC**

**Reason:** The current data file is assumed to be in ISCII or ASCII code. This occurs when DCB=OPTCD=Q is specified on the TAPEIN DD card, or when ISO/ANSI labels were detected on TAPEIN by FATAR.

**Action:** All data blocks in the file will be translated to EBCDIC for printing and processing.

**FATS039 \*\*WARNING\*\* BLOCK LENGTH GREATER THAN BLOCKSIZE OF bbbbbb**

**Reason:** A block was read whose length is greater than the blocksize “bbbbbb” of the current file (extracted from the header labels of the file, or given on the TAPEIN DD statement if labels are not processed).

**Action:** None. Unless dropped or modified, the block will be written to TAPEOUT as read.

**FATS040 TAPEIN IS NOT LABELED – LABELS=NO ASSUMED**

**Reason:** The LABELS=YES parameter was specified or defaulted, but FATAR did not find valid IBM or ISO/ANSI labels in the first file read.

**Action:** FATAR will proceed as if LABELS=NO were specified.

**FATS042 TAPEIN TRAILER BLOCK COUNT OF bbbbbb IS INCORRECT**

**Reason:** The EOF1 or EOVI trailer label on a labeled TAPEIN file contained a block count “bbbbbb” which did not match the number of blocks actually read in the file by FATAR. The actual block count is found in the “end of file” message preceding this message.

**Action:** None

**FATS044 END OF OUTPUT TAPE vvvvvv – NEW VOLUME REQUESTED**

**Reason:** The end-of-tape reflector (tape indicate) was detected on TAPEOUT volume “vvvvvv”.

**Action:** EOVI labels are written on “vvvvvv” (if labeled), and a new output volume is requested.

**FATS046 \* \* I/O ERROR-> ECB=eeeeeeee ECB rc description**

**Reason:** This is the first of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the event control block (ECB, 4 bytes) in hexadecimal along with the description associated with the ECB reason code (1st byte of ECB).

**Action:** Other messages will indicate whether FATAR can continue or must terminate, depending on the nature of the error. Appropriate manufacturer’s manuals should be consulted to determine the cause of the error. Contact INNOVATION if you need assistance.

## 40.4 CONTINUED . . .

**FATS047   \*\* I/O ERROR→ CSW=wwwwwwww wwwwwwww   *csw status desc***

**Reason:** This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the channel-status word. (CSW, 8 bytes) in hexadecimal along with the description associated with the unit and channel status bytes of the CSW (bytes 4 and 5).

**Action:** See message FATS046

**FATS048   \*\* I/O ERROR→ CCW=cccccccc cccccccc   *ccw channel cmd***

**Reason:** This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the channel-command word. (CCW, 8 bytes) in hexadecimal along with the channel command associated with the command code byte of the CCW (byte 0).

**Action:** See message FATS046

**FATS049   \*\* I/O ERROR→ SENSE ssss EXTENDED SENSE ee...ee   *error description***

**Reason:** This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the sense bytes (ssss, 2 bytes) in hexadecimal. If available, the extended sense (up to 32 bytes) is displayed. If FATAR can determine the cause from the sense and extended sense, a text description of the error is displayed.

**Action:** See message FATS046

**FATS051   RECORD rr BYTE bb LENGTH ll INVALID xxxx DATA FIELD**

**Reason:** A SCAN statement has detected data which does not meet the user-specified criteria. "rr" is the record number within the current block ("RECORD rr" will not appear if deblocking is not being done). "bb" is the location of the invalid field within the indicated record or block (relative to 1). "ll" is the length of the field. "xxxx" will be "ZONED", "PACKED" or "TYPE t" (where "t" is a table ID).

**Action:** None

**FATS052   SCAN FIELD NOT IN RECORD – SCAN TERMINATED FOR THIS RECORD**

**Reason:** A SCAN statement applied to a block or record, but the location to be scanned (P= parameter) was beyond the end of the block/record.

**Action:** SCAN not performed. Check the SCAN parameters.

**FATS053   BLOCK LENGTH ZERO – BLOCK NOT KEPT**

**Reason:** A KEEP statement applied to an input block, but the length of the data read was zero (probably due to a data check).

**Action:** The block will not be written to TAPEOUT.

## 40.4 CONTINUED . . .

**FATS054 BLOCK LENGTH CHANGED WHILE IN LOGICAL RECORD MODE –  
DEBLOCKING TERMINATED**

**Reason:** FATAR was deblocking input blocks into individual records, but a KEEP statement changed the length of a block.

**Action:** The block will not be deblocked.

**FATS056 RECORD TOO SHORT – NO DATA REPLACED**

**Reason:** A record or block is too short to contain the data specified on a REPLACE statement

**Action:** REPLACE not performed. Check the REPLACE parameters.

**FATS057 DATA RECOVERED – BLOCK NOT MODIFIED**

**Reason:** A MODIFICATION/SCAN statement with the DCK parameter applied to a data block, but the block did not have a permanent data check.

**Action:** The MODIFICATION or SCAN will not be performed.

**FATS058 RECORD *nnnnn* HAS BEEN DROPPED**

**Reason:** In response to a DROP statement with a record number indicated, record “nnnnn” has been deleted from the current block.

**Action:** The block will be “compressed” to eliminate the deleted record. If the record format is variable, the block length will be updated.

**FATS061 RECFM/LRECL NOT GIVEN – FILE ASSUMED UNBLOCKED**

**Reason:** The record format (RECFM) and logical record length (LRECL) for the current file are not available from either the TAPEIN DD statement or TAPEIN header labels.

**Action:** FATAR is unable to deblock the current file into logical records and will treat each block as a single record. If deblocking is desired, specify RECFM and LRECL on TAPEIN and re-execute FATAR.

**FATS062 DEBLOCKING ANSI SPANNED RECORDS NOT SUPPORTED – UNBLOCKED ASSUMED  
FOR THIS FILE**

**Reason:** The TAPEIN DD statement or TAPEIN header labels specifies ANSI variable-length spanned records (record format DS or DBS). FATAR does not support deblocking of ANSI spanned records.

**Action:** Unblocked (record format U) assumed.

## 40.4 CONTINUED . . .

**FATS063 BLOCK INVALID FOR RECFM/LRECL IN USE--DEBLOCKING TERMINATED**

**Reason:** Blocks in the current file were being deblocked into logical records, but the current block cannot be deblocked using the record format and logical record length provided (via the TAPEIN DD statement or TAPEIN header labels). For fixed length records, the block is not an exact multiple of the record length. For variable length records, the record descriptor words (RDW) may be invalid or indicate a length greater than the block size.

**Action:** The current block will be treated as unblocked. If the record format and length given are correct, the block is invalidly formatted; if not, specify the correct RECFM and LRECL on TAPEIN and re-execute FATAR.

**FATS064 FILE NOT ELIGIBLE FOR REBLOCKING--REBLOCKING SUPPRESSED FOR THIS FILE**

**Reason:** REBLOCK= was specified, but the current file cannot be reblocked for one of the following:

- 1) The record format is not fixed or variable
- 2) The record format is variable spanned
- 3) The blocksize is zero
- 4) The logical record length is zero (if the record format is FB or VB)
- 5) The blocksize is greater than the MAXRBLK= parameter, if specified.

**Action:** The file will be copied without reblocking.

**FATS065 RECORD LENGTH TOO LARGE FOR REBLOCKING – REBLOCKING SUPPRESSED FOR THIS FILE**

**Reason:** REBLOCK= was specified, but the logical record length of the current file is larger than the specified reblocking block size.

**Action:** The file will be copied without reblocking.

**FATS066 THIS FILE WILL BE REBLOCKED TO A BLOCKSIZE OF *nnnnn***

**Reason:** REBLOCK= was specified, and the current file can be reblocked. The new block size “nnnnn” will be equal to the reblocking block size for variable length files, or the next lower multiple of the record length for fixed length files.

**Action:** The file will be copied and reblocked.

**FATS067 INPUT DEBLOCKING ERROR WHILE REBLOCKING – FATAR TERMINATED**

**Reason:** REBLOCK= was specified, and the current file was being reblocked, but an error occurred extracting logical records from the current input block (message FATS063 will precede this message).

**Action:** Since the labels on the output tape reflect the new blocksize, FATAR will be terminated with a U0888 abend or return code of 12. Specify the correct RECFM and LRECL on TAPEIN and re-execute FATAR.

## 40.4 CONTINUED . . .

**FATS068 INPUT RECORD TOO LARGE WHILE REBLOCKING – FATAR TERMINATED**

**Reason:** REBLOCK= was specified, and the current file was being reblocked, but an input logical record was larger than the output reblocking blocksize.

**Action:** Since the labels on the output tape reflect the new blocksize, FATAR will be terminated with a U0888 abend or return code of 12. Specify a larger value for REBLOCK= and re-execute FATAR.

**FATS069 RECORD *nnnnn* SPANNING ERROR – *xxxxx* OF SEGMENT MISSING**

**Reason:** The TAPEIN JCL or tape labels specifies variable spanned records for the current file, and FATAR has detected an incomplete spanned record. "xxxxx" will be "START" if the end of a record was found without a corresponding beginning, or "END" if a new record was found before a previous spanned record was complete. "nnnnn" is the record number within the current block where the error was detected (if "xxxxx" is "END", the error is probably actually in the preceding block(s)).

**Action:** An additional FATAR run may be required to print the appropriate blocks and determine which records are truly in error. Another run using either FATAR REPLACE statements to fix the spanning flags, or DROP statements to delete the partial records may be used to fix the tape.

**FATS070 CONTROL CARD TABLE SIZE IS *n* BYTES**

**Reason:** This message is always printed to document "n", the size of the FATAR control statement storage table. This size defaults to 32768 but can be increased by the SIZE= parameter in the FATAR EXEC JCL statement PARM=.

**Action:** If this FATAR execution receives message FATS010 reason G, increase the table size.

**FATS071 TAPE BUFFER SIZE IS *nnnnnn* BYTES**

**Reason:** This message is always printed to document "n", the size of the FATAR TAPEIN read buffer. The default size is 262144 or 256K bytes but can be reduced by use of the BLKSIZE= parameter.

**Action:** Any blocks on TAPEIN exceeding "n"-1 in length will cause an error. If FATAR is creating an output tape, it will terminate, otherwise it will continue (see message FATS036).

**FATS072 TAPEIN DATA WILL BE COPIED TO TAPEOUT**

**Reason:** The TAPEOUT DD statement is present.

**Action:** All files and data on TAPEIN will be copied to TAPEOUT unless "DROPPED" by data checks or FATAR control statements.



## 40.4 CONTINUED . . .

**FATS074 \*\* WARNING \*\* CONTROL CARD NUMBER N WAS NEVER REFERENCED OR REPLACE-  
VERIFY NEVER SATISFIED**

**Reason:** This message is issued at FATAR termination to warn of control statements which have never been acted upon (possibly due to user error). Either the file \$NL or block \$NL on the statement was never encountered on TAPEIN, or, for a REPLACE statement with a V= or S= parameter, the VERIFY or SCAN never caused any data to be replaced.

**Action:** Review the control statements to be sure that they were coded correctly. Review the FATAR output to be sure that the desired files and blocks were processed.

**FATS075 SPECIAL EXPIRATION OF yydd1 DETECTED, TAPEOUT EXPDT=yydd2**

**Reason:** FATAR read labels containing an expiration date "yydd1" recognized as having a special purpose, i.e., 98000 or 99000.

**Action:** If copying to TAPEOUT, the output file will have expiration "yydd2" (98000 is changed to 00000 but some tape management systems will change this to a default retention).

**FATS076 TAPE MANAGEMENT text**

**Reason:** Displays information about the FATAR Tape Management interface. If no supported tape management system was detected or an interface error occurred, this is indicated. Otherwise it will identify the type and level of the tape management system, and information about the input tape extracted from that TMS.

**Action:** None, unless an error occurs. If necessary, contact Innovation Technical Support.

**FATS077 EOT ON TAPEOUT BUT FATAR WILL ATTEMPT TO WRITE nnn MORE BLOCKS**

**Reason:** During an image copy (LABELS=NO), FATAR detected the logical End-of-Tape (EOT) on the output tape, and EOTBLOCKS= was specified or defaulted.

**Action:** Since an image copy must put all the input data on a single output volume, FATAR will attempt to write "nnn" additional blocks beyond logical EOT.

**FATS078 EOT ON TAPEOUT, FATAR WROTE nnn MORE BLOCKS BUT COPY NOT COMPLETE**

**Reason:** During an image copy (LABELS=NO), FATAR wrote extra blocks beyond logical EOT, but the allowable block count was reached before all input data was copied.

**Action:** Since an image copy must put all the input data on a single output volume, this copy is not complete. If "nnn" is less than 999, you can attempt to complete it by increasing the EOTBLOCKS= value, or using a longer output tape.

**FATS079 PHYSICAL EOT ON TAPEOUT AFTER nnn EXTRA BLOCKS WRITTEN, COPY INCOMPLETE**

**Reason:** During an image copy (LABELS=IMAGE), FATAR wrote extra blocks beyond logical EOT, but physical EOT was detected (on a cartridge drive) before all input data was copied.

**Action:** Since an image copy must put all the input data on a single output volume, this copy is not complete. You can attempt to complete it by using a longer output tape.

## 40.4 CONTINUED . . .

**FATS080 UNABLE TO LOAD CHARTAB=*table* PRINT TRANSLATE TABLE.  
DEFAULT "FATCHRTB" WILL BE USED.**

**Reason:** CHARTAB= was specified with the name of a load module containing an alternate print table, but FATAR was unable to successfully LOAD that module.

**Action:** Check the job log for IBM messages indicating why the module could not be loaded. For this run the standard FATAR translate table is used.

**FATS081 TAPE LABELS IN ISO/ANSI FORMAT – TRANSLATED TO EBCDIC FOR PRINTING ONLY**

**Reason:** ANSI labels were detected on the input tape. They are translated to EBCDIC for printing purposes only.

**FATS082 READ BUFFER NOT AVAILABLE – USE DUMP=YES OPTION**

**Reason:** This is an internal error. Contact Innovation for assistance.

**FATS083 TAPE REWOUND TO LOAD POINT FOR IMAGE COPY**

**Reason:** An image copy was requested (LABELS=IMAGE) but the TAPEIN JCL specified a file number (LABEL=n) greater than 1.

**Action:** Since OPEN positioned the tape to the specified file, FATAR rewound the tape to the beginning (load point) in preparation for copying the entire tape.

**FATS084 \*\*WARNING\*\* MORE THAN ONE INPUT VOLUME FOUND DURING IMAGE COPY.  
COPY STOPPED AT END OF FIRST VOLUME**

**Reason:** An image copy was requested (LABELS=IMAGE) but the TAPEIN JCL specified more than one volume serial. Image copy can process only one volume at a time.

**Action:** The additional volume serials were ignored. If they must also be copied, you must submit separate LABELS=IMAGE steps for each one.

**FATS085 \*\*WARNING\*\* NUMFILES SET TO 0 FOR IMAGE COPY**

**Reason:** An image copy was requested (LABELS=IMAGE) but NUMFILES= (NF=) was specified with a value other than 0..

**Action:** Since image copy must copy the entire tape, NUMFILES=0 was forced.

**FATS086 \*\*WARNING\*\* NO TAPEOUT DD FOUND WHEN LABELS=IMAGE SPECIFIED**

**Reason:** An image copy was requested (LABELS=IMAGE) but no TAPEOUT DD was provided to specify the output tape.

**Action:** Processing will continue, allowing you to "dry run" the image copy, but the step must be rerun with a TAPEOUT DD statement to actually copy the tape.

## 40.4 CONTINUED . . .

**FATS087 \*\*WARNING\*\* CATALOG OF OUTPUT TAPE FAILED. COMP=*comp* DSN=*dsn***

**Reason:** Cataloging of the indicated output dataset was requested by the CAT=YES or CAT=RECAT parameters, but the attempt to catalog the indicated dataset failed. "comp" is the return code from CAMLST CATALOG, which will usually be 0008. For CAT=YES this may indicate that the output dataset name is already cataloged. Another common cause is attempting to catalog into the master catalog, usually because you have used a output dataset name which is not assigned to a user catalog. FATAR will complete processing but the indicate output dataset names will not be cataloged.

**Action:** You can catalog the output dataset name manually without rerunning FATAR, using the IDCAMS statement DEFINE NONVSAM. If you want to recatalog the output name to the output tape, rerun the step using CAT=RECAT instead of CAT=YES. If the output name is not valid, use the RENAME statement to change the name to one that can be cataloged.

**FATS088 \*\*WARNING\*\* CATALOGING REQUESTED BUT FULL NAME NOT AVAILABLE. DSN=*dsn***

**Reason:** Cataloging of the indicated output dataset was requested by the CAT=YES or CAT=RECAT parameters, but was not attempted because FATAR does not know the full 44-character name of the dataset. IBM records only the last 17 characters of the original dataset name in the tape labels, which is not valid for cataloging. FATAR can attempt cataloging if:

- the FATAR tape management interface is enabled and has provided the full dataset name from tape management records
- the name in the labels is 16 characters or less
- you have provided the output dataset name on a RENAME statement with the NEWN= parameter.

FATAR will complete processing but the indicate output dataset names will not be cataloged.

**Action:** You can rerun the FATAR step providing RENAME statements with NEWN= to provide the output dataset name. If the tape is recorded in your tape management system but you have not enabled the FATAR tape management interface, you can do so (See Section 90.3).

**FATS089 \*\*WARNING\*\* NEWI= SPECIFIED BUT FULL NAME NOT AVAILABLE. NEWI=*mask***

**Reason:** A RENAME statement with the NEWI= operand applies to this file, but the full dataset name of the input dataset was not available. The full name is available if the original name is 16 characters long or less, if the full name was provided on the TAPEIN DD statement, or if you have enabled the FATAR tape management interface.

**Action:** The output file will be created with the truncated name derived from the input tape labels. You may want to rerun the copy specifying a full output dataset name with NEWN= instead of NEWI=.

**FATS090 \*\*WARNING\*\* NEWI= RESULTED IN INVALID OUTPUT DATASET NAME. NEWI=*mask***

**Reason:** A RENAME statement with the NEWI= operand applies to this file, but when the mask was applied to the input dataset name, it resulted in an output dataset name which does not meet IBM standards.

**Action:** The output dataset will be created with the input dataset name, ignoring the RENAME statement. You may want to rerun the copy specifying a NEWI= which results in a valid name, or specify a NEWN= to provide a complete output name.

## 40.4 CONTINUED . . .

**FATS091 \*\*WARNING\*\* NEXTVOL=TMS SPECIFIED BUT TMS INFORMATION NOT AVAILABLE.**

**Reason:** NEXTVOL=TMS was specified, requesting that FATAR get information about the additional input tape volumes related to the first input tape from the FATAR tape management interface, but no tape management information was available for this tape. Either the tape management interface is not enabled (See Section 90.3) or the input tape is not recorded in the tape management data base.

**Action:** If additional volume serials were specified on the TAPEIN DD statement, they will be mounted. Otherwise, FATAR will terminate normally at the end of the input tape.

**FATS092 OUTPUT DATASET RENAMED TO DSN=*dsname***

**Reason:** A RENAME statement was supplied which applies to this tape file. The actual name of the output dataset is displayed.

**FATS093 OUTPUT DATASET RENAMED AND CATALOGED. FILE=*ffff*,UNIT=*xxxxxxxx*,DSN=*dsname***

**Reason:** A RENAME statement was supplied which applies to this tape file and CAT= was also specified. The actual name of the output dataset is displayed; this dataset was cataloged with the file number and device type (in hex) shown.

**FATS094 \*\*WARNING\*\* CATALOGING REQUESTED BUT NEW NAME WAS NOT SPECIFIED.**

**Reason:** CAT= was specified, requesting that this output dataset be cataloged, but the full name of the input dataset was not available and no RENAME NEWN= was provided with an output dataset name. The full name is available if the original name is 16 characters long or less, if the full name was provided on the TAPEIN DD statement, or if you have enabled the FATAR tape management interface.

**Action:** The output file will be created with the truncated name derived from the input tape labels but this name was not cataloged.

**FATS095 OUTPUT DATASET CATALOGED. FILE=*ffff*,UNIT=*xxxxxxxx*,DSN=*dsname***

**Reason:** CAT= was specified, requesting that this output dataset be cataloged, and the current output dataset was cataloged with the file number and device type (in hex) shown.

**FATS096 END OF DATA MARK ENCOUNTERED. BLOCK ID=*xxxxxxxx***

**Reason:** FATAR encountered an End-of-Data (EOD) hardware mark on an input tape. An EOD mark is written on every cartridge (except 3480-format) after the trailer labels of the last file on the tape, so this indicates that your FATAR options requested FATAR to read beyond that last file. "xxxxxxxx" is the hardware block ID of the EOD mark.

**Action:** FATAR will attempt to position past the EOD mark and continue. However, the results may vary depending on the type of tape and the vendor of the hardware; sometimes FATAR may be unable to move past the EOD and other times it may need to skip over some data past the EOD.

**40.4 CONTINUED . . .****FATS097 LABELS=ONLY SPECIFIED. DATA FILE WILL BE SKIPPED.**

**Reason:** LABELS=ONLY was specified on the ANALYZE/COPY statement. FATAR will read and format tape labels, but will use a FSF (Forward Space File) CCW to rapidly skip over data files. No meaningful statistics about the contents of the data files will be printed.

**FATS098 LABELS=ONLY SPECIFIED. NO TOTALS WILL BE PRODUCED FOR THIS RUN.**

**Reason:** LABELS=ONLY was specified on the ANALYZE/COPY statement. FATAR will read and format tape labels, but will use a FSF (Forward Space File) CCW to rapidly skip over data files. FATAR totals are meaningless without information about the data files, so they are not printed.

**FATS100 (INSTRUCTION LINE)**

**Reason:** If PARM=I is present on the FATS EXEC JCL statement, this message will be printed repeatedly on SYSPRINT with the internal FATS instructions.

## 40.4 CONTINUED . . .

**FATS102 CONTROL CARD ERROR – REASON=x**

**Reason:** A syntax or usage error has been detected in the control statement printed immediately preceding the message. Only one FATS102 message will be printed for each input statement (multiple errors on a statement will not be diagnosed). “x” is a letter or number indicating the type of error detected:

- 1 – An option parameter is not followed by a blank or comma.
- 2 – Operation keyword is invalid or the tape number in parentheses is not 1 through 9.
- 3 – Tape number has been previously used on another operation statement.
- 4 – An option parameter is invalid.
- 5 – Error in the MODE= parameter. Value is not 2 hexadecimal digits or is not one of the valid values.
- 6 – Error in the RETRY= parameter. Must be RETRY=h or RETRY=(l,h). “h” must be 1 to 99. “l” must be 0 to “h”.
- 7 – Error in the numeric value of a keyword parameter. Invalid numeric digits.
- 8 – Error in the BPI= parameter, BPI=0.
- 9 – Error in the LABEL= parameter, invalid characters in the label value.
- A – Expected continuation statement not received (previous statement ended in a comma).
- B – No ending apostrophe in OWNERID= parameter.
- C – Error in the THRESHOLD= parameter, greater than 32767.
- D – Label operation statement missing LABEL= or VOL= parameter.
- E – Error in BLKSIZE= parameter, must be less than 65536 and more than twice the tape density (BPI), except for cartridges.
- F – Error in VOL= parameter. Volume serial longer than 6 characters.
- G – Error in VOL= parameter. More than 682 volume serials specified.
- H – VOL= parameter specified on a default statement.
- I – VOLINCR= parameter specified without VOL= parameter.
- J – Blank control statement.
- K – Error in VOLINCR= parameter. The last volume serial in the VOL= parameter could not be incremented due to insufficient trailing numeric digits.
- L – Error in ENDAFTER= parameter (for FATAR execution).
- M – Invalid numeric value for keyword. Value was zero or greater than 32767.
- N – An option parameter was specified which is invalid for the control statement on which it was found. For instance, MODIFY=YES was specified on a read statement, but it is only valid on a DEFAULT statement.

**Action:** FATS will abend with a U0502 abend code after all control statements have been scanned.

**FATS104 ONE OR MORE ERRORS DETECTED — FATS TERMINATED**

**Reason:** Message FATS102 was issued for one or more control statement errors.

**Action:** FATS will abend with a U0502 abend code. Correct the errors and re-execute.

**FATS105 NO INPUT CARDS – FATS TERMINATED**

**Reason:** No control statements other than DEFAULT and comments were present in the input. At least one operation statement must be present.

**Action:** FATS will abend with a U0502 abend code.

## FATS106 DD=SYSPRINx MISSING – REQUIRED FOR ANALYZE

**Action:** FATS willabend with a U0502abend code.

FATS108 NO SUMMPRT DD PROVIDED-SUMMARY REPORT WILL GO TO SYSPRINT

**Action:** FATS will write summary report data to SYSPRINx if present for TAPEx or to SYSPRINT

**Action:** Other operation statements (if present) will be processed. Check the TAPEX DD statement.

**Action:** FATS will parse the control statement.

**Action:** FATS will perform the requested functions.

**Action:** FATS will terminate with a U0047 ABEND.

**Action:** The program will terminate with a U0913 abend. If you require the failing function or option, contact your security administrator. See Sections 11.8 (FATS) and 21.6 (FATAR) for information on the security options.

## 40.4 CONTINUED . . .

**FATS114 PROGRAM TERMINATED. USER NOT AUTHORIZED TO READ DSN=*dsname***

**Reason:** Your security system has indicated that you are not authorized to read a file on the input tape with dataset name "*dsname*".

**Action:** The program will terminate with a U0913 abend. If you require access to this dataset, contact your security administrator. See Section 21.6 for information on the FATAR dataset security.

**FATS115 UNLABELED TAPE USED AS INPUT WHEN LABELS=ONLY WAS SPECIFIED. RUN TERMINATED.**

**Reason:** LABELS=ONLY is supported only on a labeled tape.

**Action:** The program will terminate.

**FATS116 UNABLE TO ACCESS DATA PAST END-OF-DATA MARK. RUN TERMINATED.**

**Reason:** All cartridge tapes (except 3480) put a hardware EOD (End-of-Data) mark after the last block or tape mark written to the tape. This message indicates that FATAR was unable to position past the EOD mark. This can occur when:

- 1) LABELS=EOD was specified, to recover data past the EOD.
- 2) LABELS=NO and NUMFILES=*n* was used to request processing past the EOD.

**Action:** The program will terminate

**FATS117 ANALYSIS TERMINATED AT END-OF-DATA MARK**

**Reason:** On a cartridge tape (except for 3480), FATS encountered a EOD (End-of-Data) mark which follows a tape mark.

**Action:** Processing of the input tape is completed.

**FATS118 UNABLE TO LOAD *name* ASCII TRANSLATE TABLE. DEFAULT IBM TABLE WILL BE USED**

**Reason:** The ASCII=*name* operand was specified, but a load module called "*name*" could not be successfully loaded, so the default IBM ASCII translate table will be used.

**Action:** Make sure that the load module name of the ASCII translate table was properly specified, and that it is in a load module library accessible by this FATAR step.

**FATS119 COPY OR SIMULATE SPECIFIED, BUT NO SELECT CARDS WERE PROCESSED. JOB TERMINATED.**

**Reason:** FATSCOPY was executed but no SELECT statements were input.

**Action:** Correct the control statements and re-submit.

**FATS120 DSN=*dsname* ON VOLUME=*volser* WAS SELECTED FOR COPYING.**

**Reason:** FATSCOPY selected the indicated tape dataset on the indicated tape volume from the system catalog for copying.



## 40.4 CONTINUED . . .

**FATS121 NO DATASETS WERE SELECTED FOR COPYING. JOB TERMINATED.**

**Reason:** The SELECT/EXCLUDE statements specified for FATSCOPY did not result in any datasets being selected.

**Action:** If the statements were specified incorrectly, correct and re-submit. If datasets should have been selected, check the statements carefully for errors; if necessary, contact Innovation for assistance.

**FATS122 FATSCOPY SIMULATE SUCCESSFULLY COMPLETED. *nnnn* DATASETS WERE SELECTED FOR COPYING.**

**Reason:** FATSCOPY was executed with a SIM (SIMULATE) statement. If it had been executed with a COPY statement, "*nnnn*" datasets would have been copied. The selected datasets are listed in the SELRPT and COPYRPT reports.

**FATS123 LOAD OF RMM API FAILED. RMM DATA FOR DSN=*dsname* WILL NOT BE DELETED.**

**Reason:** Your tape management system is apparently DFSMSrmm (RMM) but the RMM API (application program interface) could not be used by FATSCOPY.

**Action:** Contact Innovation for assistance.

**FATS124 SELRPT DD STATEMENT MISSING. DATASET SELECTION REPORT WILL NOT BE GENERATED.**

**Reason:** You omitted the SELRPT DD statement in a FATSCOPY step. The report of datasets selected from your catalogs will not be produced, but FATSCOPY will continue to execute.

**Action:** If you wanted the report of selected datasets, add the SELRPT DD to your step; it is usually a SYSOUT=\* dataset.

**FATS125 COPYRPT DD STATEMENT MISSING. DATASET COPY/SIM REPORT WILL NOT BE GENERATED.**

**Reason:** You omitted the COPYRPT DD statement in a FATSCOPY step. The report of datasets written to output tapes will not be produced but FATSCOPY will continue to execute. If this is a SIMULATE (SIM), this report would have been a list of datasets which would have been written to an output tape.

**Action:** If you wanted the report of output datasets, add the COPYRPT DD to your step; it is usually a SYSOUT=\* dataset.

**FATS126 MAXIMUM NUMBER OF INPUT TAPE FILES EXCEEDED. SELECT STATEMENT NOT PROCESSED.**

**Reason:** The preceding SELECT statement was ignored because the maximum number of files to be copied in this FATSCOPY step (specified by MAXTOTFILE=) has been exceeded.

**Action:** Include the SELECT statement(s) in another FATSCOPY step to copy those datasets.

## 40.4 CONTINUED . . .

**FATS127 NO TMS INFORMATION FOUND. TMS INFO WILL NOT BE COPIED FOR DSN=*dsname*.**

**Reason:** The indicated dataset was selected from your system catalogs by FATSCOPY, but no tape management information was found for the dataset.

**Action:** The dataset will be copied but no TMS information will be updated for this dataset.

**FATS128 LOAD OF RMM API FAILED. RMM DATA FOR DSN=*dsname* WILL NOT BE COPIED.**

**Reason:** Your tape management system is apparently DFSMSrmm (RMM) but the RMM API (application program interface) could not be used by FATSCOPY.

**Action:** Contact Innovation for assistance.

**FATS129 DYNAMIC ALLOCATION FAILED FOR VOL=*volser* DEVTYPE=*devtype*. RC=*rrrr*, REAS=*cccc*.**

**Reason:** FATSCOPY issued a dynamic allocation for an input tape file to copy on the volume serial and tape device type shown, but the allocation failed with the indicated return code and reason code.

**Action:** The tape file will be bypassed. For OS/390, these codes are documented in the IBM manual *Authorized Assembler Services Guide*. They can also be found in Appendix A of the ISPF online HELP. Frequently encountered "nnnn" values include:

**0210** requested dataset not available (e.g., another job had the tape file allocated with DISP=OLD).

**021C** invalid unit name used (e.g., FATSCOPYH is trying to allocate a 3480 on a system without 3480s).

**0220** requested volume not available (e.g., another job is using a tape volume required by FATSCOPY).

**FATS130 DYNAMIC DEALLOCATION FAILED FOR VOL=*volser* DEVTYPE=*devtype*. RC=*rrrr*, REAS=*cccc*.**

**Reason:** FATSCOPY issued a dynamic de-allocation for an input tape file to copy on the volume serial and tape device type shown, but the de-allocation failed with the indicated return code and reason code.

**Action:** FATSCOPY will terminate. For OS/390, these codes are documented in the IBM manual *Authorized Assembler Services Guide*. They can also be found in Appendix A of the ISPF online HELP.

**FATS131 EXPIRATION DATE FOR INPUT VOLUME=*volser* HAS BEEN CHANGED TO TODAY'S DATE.**

**Reason:** The user specified TMSINPUT=SCRATCH and the expiration date for the volume has been changed to today's date so that the volume will expire the next time tape management maintenance runs.

**FATS132 TAPE MGMT INFO FOR DSN=*dsname* ON VOL=*output* WAS COPIED FROM VOL=*input***

**Reason:** FATSCOPY has copied the appropriate information, such as expiration date, creating job, etc. from the tape management record for the input volume to the tape management record for the output volume.

## 40.4 CONTINUED . . .

**FATS133 EXPIRATION DATE FOR VOLUME=*volume* CHANGED TO *date***

**Reason:** FATSCOPY has changed the expiration date for the output volume to the date specified. This is the latest date for any dataset on the volume. (This message may only apply to RMM.)

**FATS134 UNABLE TO CHANGE EXPIRATION DATE FOR OUTPUT VOLUME=*volser*  
RC=*return code*,REASON=*reason code***

**Reason:** FATSCOPY's attempt to update the tape management expiration date for the output volume failed.

**FATS135 UNABLE TO EXPIRE INPUT VOLUME=*volser* RC=*return code*,REASON=*reason code***

**Reason:** FATSCOPY tried to expire the input volume but was unable to do so.

**FATS136 UNABLE TO DELETE TMS DATASET INFO FOR OUTPUT DSN=*dsname* VOLUME=*volser***

**Reason:** While copying a dataset, FATSCOPY reached the end of tape on the OUTPUT. At the time it reached end of tape, the number of bytes written to TAPEOUT was less than the value specified in the LASTFILE= parameter. FATSCOPY attempted to delete the tape management information for the output dataset, but failed to do so.

**FATS137 NO TMS INFO FOR DSN=*dsname*. UNABLE TO EXPIRE INPUT TAPE**

**Reason:** TMSINPUT=SCRATCH was specified but there was no tape management information for the input dataset.

**FATS138 FATSCOPY UNABLE TO DELETE RMM INFO FOR VOLUME=*input volser*.  
ALL DATASETS ON INPUT NOT SELECTED FOR COPYING**

**Reason:** TMSINPUT =SCRATCH was specified, and RMM was active. With RMM, deleting information for a dataset on the volume will also delete the tape management information for any dataset on the volume with a higher file sequence number. FATSCOPY determined that all the datasets on the volume were not selected for copying and bypassed expiring the volume.

**FATS139 xxx UPDATE FAILED FOR VOLUME=*volser*,DSN=*dsname*,  
RC=*return code*,REAS=*reason code*.**

**Reason:** TMSDATA=COPY was specified, but FATSCOPY's attempt to update the tape management information for the volume failed.

**FATS140 COPY FUNCTION SPECIFIED BUT NO TAPEOUT DD FOUND. JOB TERMINATED.**

**Reason:** The COPY statement requires a TAPEOUT DD statement.

**FATS141 TMS NOT ACTIVE. TMS DATA WILL NOT BE COPIED. DSN=*dsname*,VOL=*volser*.**

**Reason:** The tape management system expected was not active, so FATSCOPY will not attempt to copy tape management data relating to the datasets it is copying.

## 40.4 CONTINUED . . .

**FATS142 ERROR OCCURRED READING TMC FOR xxxxxx VOL=volser. TMS DATA WILL NOT BE COPIED.**

**Reason:** An error occurred while reading tape management data for the input tape (xxxxxx=INPUT) or the output tape (xxxxxx=OUTPUT). The tape management data for the file being copied will not be updated on the output tape's records.

**FATS143 ERROR OCCURRED UPDATING TMC FOR DSN=dsname VOL=volser.**

**Reason:** An error occurred while updating tape management data for the output dataset. The tape management data for the file being copied will not be updated on the output tape's records.

**FATS144 ERROR OCCURRED DEQUEUEING TMC FOR DSN=dsname VOL=volser.**

**Reason:** An error occurred while updating tape management data for the output dataset. The tape management data for the file being copied will not be updated on the output tape's records.

**FATS145 DSN=dsname ON VOL=volser BYPASSED – MISSING VOLS OF MULTI-VOL DSN.**

**Reason:** When processing a dataset that had met all the selection criteria, FATSCOPY determined that some volumes of a multi-volume set had not met the selection criteria. FATSCOPY will not copy the dataset.

**FATS146 UNSUPPORTED TAPE MGMT SYSTEM. UNABLE TO UPDATE TAPE MANAGEMENT INFORMATION**

**Reason:** TMSDATA=COPY or TMSINPUT=SCRATCH was specified, but FATSCOPY didn't find either CA-1 or RMM, the only tape management systems currently supported.

**FATS147 TMC OPEN FAILED. EXPIRATION DATE FOR INPUT VOL=volser WILL NOT BE CHANGED.**

**Reason:** An error occurred when FATSCOPY tried to obtain the expiration date for the input volume prior to changing the expiration date to the current date. The expiration date will not be changed.

**FATS148 TMC OPEN FAILED. TMS DATA CAN NOT BE COPIED. DSN=dsname,VOL=volser.**

**Reason:** An error occurred when FATSCOPY tried to open the TMC to obtain tape management information for the input volume to update tape management information for the output volume. The TMS information will not be copied to the output.

**FATS149 TMS ERROR CODES: RC=return code,REAS=reason code,  
SEC RC=security return code,SEC REAS=security reason code.**

**Reason:** Issued after an error reading, updating, or dequeuing the TMC occurs. Contains CA-1 abend codes, reason codes, and security return and reason codes.

**FATS150 TMS NOT ACTIVE. INPUT TAPE WILL NOT BE EXPIRED. VOL=volser**

**Reason:** TMSINPUT=SCRATCH was specified, but TMS was not active.

## 40.4 CONTINUED . . .

**FATS151 ERROR OCCURRED xxxxxx TMC FOR INPUT VOL=volser. TAPE WILL NOT BE EXPIRED.**

**Reason:** An error occurred when FATSCOPY tried to obtain the expiration date for the input volume prior to changing the expiration date to the current date (xxxxxx=READING) or when FATSCOPY tried to update TMS with the new expiration date (xxxxxx=WRITING). The expiration date will not be changed.

**FATS152 EOT ON TAPEOUT BEFORE LASTFILE VALUE REACHED. DSN=outdsn,VOL=outvol**

**Reason:** The physical end-of-tape was reached before FATSCOPY copied the number of bytes specified in the LASTFILE= keyword. FATSCOPY will call for a new output tape, determine the tape the input file started on, and begin copying the input file from the beginning. If tape management is active, FATSCOPY will expire the partially copied tape on the output tape.

**FATS153 PARTIALLY COPIED DSN=outdsn ON OUTPUT VOL=outvol WAS EXPIRED**

**Reason:** FATSCOPY successfully expired the partially copied dataset on the output tape in tape management.

**FATS154 NO ACTIVE TAPE MANAGEMENT. DSN=outdsn, VOL=outvol WILL NOT BE EXPIRED.**

**Reason:** FATSCOPY attempted to expire a dataset in tape management that had been partially copied, but no active tape management system was active.

**FATS155 UNABLE TO FIT DSN=outdsn ON ONE VOLUME. LASTFILE= VALUE WILL BE IGNORED.**

**Reason:** After calling for a new output volume for a dataset that wouldn't fit on the output volume, FATSCOPY again arrived at the end of the output tape without exceeding the LASTFILE= amount. FATSCOPY will call for a new output without starting to copy the input from the beginning.

**FATS156 EXPIRATION DATE FOR DSN=outdsn, VOL=outvol WAS SET TO TODAY'S DATE**

**Reason:** After copying a dataset, FATSCOPY expired the DSNB record for the dataset in CA-1 (TMS).

**FATS157 EXPIRATION DATE FOR INPUT VOLUME=volser WAS RESET TO DATE=date**

**Reason:** When FATSCOPY has copied the first dataset on a volume containing multiple files and the tape management system is CA-1, FATSCOPY will reset the expiration date for the volume to the highest expiration date of any unexpired dataset on the volume. If it copies any other files after copying the first file, FATSCOPY will again reset the volume to the highest expiration date of any remaining datasets. If all the datasets on the volume are successfully copied, FATSCOPY will expire the volume at that time.

**FATS158 TMC OPEN FAILED. DSN=dsname, VOL=volser WILL NOT BE EXPIRED**

**Reason:** OPEN of the CA-1 TMC for update by FATSCOPY failed and it will not be able to expire the dataset. This message will be followed by a FATS149 containing the error codes returned by TMS for the failed OPEN. This may be due to security checks in CA-1; check the CA-1 documentation for information on security protection of the TMC and the YSVCUNCD resource.

## 40.4 CONTINUED . . .

**FATS159 NO DATASETS COPIED DURING FATSCOPY RUN. CHECK SYSPRINT FOR ERROR MESSAGES**

**Reason:** FATSCOPY ended with no datasets being copied, even though datasets were selected for copying. The SYSPRINT report will contain error messages to show why no datasets were copied.

**FATS160 OPEN FAILED FOR DSN=*dsn*,VOL=*volser*,ABEND=*abend code***

**Reason:** FATSCOPY tried to open an input dataset but OPEN abended with the ABEND code shown. The joblog may contain IBM messages providing more detail on the error.

**Action:** The input dataset will be skipped and FATSCOPY will continue to copy other datasets, if any are left.

**FATS161 EXPIRATION FOR DSN=*dsn*,VOL=*volser* WILL BE RESET BY RMM HOUSEKEEPING**

**Reason:** While attempting to update the RMM expiration date for a dataset it just copied, FATSCOPY found that the datasets expiration was controlled by a RMM Vital Record Specification. The next time RMM housekeeping runs, RMM will set the expiration for the output dataset according to the VRS.

**FATS162 MAXIMUM NUMBER OF FILES SELECTED FOR COPYING. CATALOG SEARCH STOPPED**

**Reason:** While searching the catalog for datasets to copy, FATSCOPY exceeded the maximum number of data specified by the MAXTOTFILE= operand (default: 1000).

**Action:** FATSCOPY will stop selecting datasets and will process the datasets it has selected. Rerun FATSCOPY to select the remaining datasets, and increase MAXTOTFILE= if needed.

**FATS163 CATALOGING BYPASSED FOR UNCATALOGED INPUT DATASET**

**Reason:** CAT=ONLY was specified (or defaulted in FATSCOPY) and an input dataset was not cataloged. This could mean that the input dataset was not in the catalog at all, or that it was cataloged but it pointed to a volume or file number different from the input dataset.

**Action:** The output file will not be cataloged.

**FATS164 FATSCOPY UNABLE TO OBTAIN LAST USED INFO FOR VOLUME=*iiiiii***

**Reason:** TMSDATA=COPY was specified, so FATSCOPY attempted to get the tape management information about the last use of the input tape "iiiiii" but was unable to do so.

**Action:** The last use information on the output tape will not be updated; it will show the FATSCOPY job as the last user.

**FATS165 LAST USED INFO FOR VOLUME=oooooo COPIED FROM VOLUME=*iiiiii***

**Reason:** TMSDATA=COPY was specified, so FATSCOPY copied the tape management information about the last use of the input tape "iiiiii" to the output tape "oooooo"

## 40.4 CONTINUED . . .

**FATS201 TAPEX MAJOR ERROR ON TAPE  
ANALYZE TERMINATED FOR VOL=vvvvvv**

**Reason:** The first format of the message will be issued whenever unrecoverable I/O errors (other than data checks) occur on the indicated tape drive; it will be followed by message FATS202. The second format of the message appears when a threshold of major errors is exceeded or when the error is too severe to continue (such as equipment check or intervention required).

**Action:** If the second form of the message is issued, further processing of the tape volume "vvvvvv" is halted, and FATS will eventually abend with a U0888 abend code to indicate that major errors occurred.

**FATS202 IOB SENSE ECB CSW CCW ADDRESS DCB FAILING CCW  
*data corresponding to the above labels*  
FULL SENSE=ff....ff**

**Reason:** This message is issued after message FATS201 when major I/O errors occur. At least the first two lines will be printed, the first containing the titles shown and the second the corresponding diagnostic data in hexadecimal. If the complete sense data produced by the device is available, it will be printed in the third line.

**Action:** Major errors may indicate severe problems with the tape being processed or problems with the tape unit being used. Appropriate manufacturer's manuals should be consulted to determine the cause of the error. Contact INNOVATION if you need assistance.

**FATS203** A line of the detail report indicating a tape mark read.

**FATS204** A line of the detail report indicating a data check.

**FATS205** A line of the detail report indicating PRINT threshold met.

**FATS206** A line of the detail report showing the compaction achieved for the current line.

**FATS207** A line of the detail report indicating operator cancelled, maximum errors exceeded, or end-of-tape tape indicate).

**FATS208** A line of the detail report indicating cleaning action.

**FATS209** A line of the detail report indicating that an IDRC compacted file was sensed.

**FATS210 FATS OPEN/CLOSE SUBTASK ABEND ON DD=TAPEX COMP CODE = Ssss Uuuuu**

**Reason:** FATS sub-task which handles OPEN, CLOSE, and FATAR has abended with a system (sss) or user (uuuu) abend.

**Action:** Use of the tape drive on TAPEX will be terminated. If no other TAPEX DD statements are active, FATS will terminate.

**FATS211 OPERATOR CANCELLED ANALYSIS ON DD=TAPEn**

Processing of this tape was cancelled by the operator by a reply of "KEOJ" or "Kuuu" to the FATSW02 console message, or "TERM" or "SKIP" to the FATSW08 console message.

**FATS300** A line of the summary report giving DD name, tape unit address, volume serial, operation performed, termination status, tape length, and data check summary by retry count.

**FATS301 END OF REPORT**

**Reason:** Printed at the end of all detail and summary reports.

## 40.4 CONTINUED . . .

**FATS345 FILTER ERROR REASON=reason – ENTRY=entry**

**Reason:** SELECT CATDSN=filter was used to select entries from the system catalogs, and an error occurred. "entry" indicates the catalog name or the catalog entry on which the error occurred. "reason" indicates the error:

- 1 — NO CATALOGED ENTRIES FOUND** — the filter did not select any entries from the catalogs.
- 2 — CATLG ERROR COMP=cccc CODE=reason** — The IBM catalog SVC returned return code "xxxx" with reason code "reason". These codes can be found under message IDC3009I in IBM message manuals.
- 3 — CATLG NAME FPL ADDR MISSING** — internal error.
- 4 — CATLG VOLSER FPL ADDR MISSING** — internal error.
- 5 — CATLG VOLSER 0 OR MORE THAN 20** — a catalog entry was selected that had either 0 volsers or more than 20 volsers in a CVOL (SYSCTLG) catalog.
- 7 — CATLG ENTYPE FPL ADDR MISSING** — internal error.
- A — NAME LAST CHARACTER HIGH VALUES** — internal error.
- D — CANNOT END IN A PERIOD** — The filter cannot end in a period. See Section 80 for filter rules.
- E — INVALID GENERATION NUMBER** — The filter ends in an invalid GDG relative generation number.
- F — SEQUENCE ERROR IN CATALOG** — A VSAM sequence error was encountered reading an ICF catalog, indicating a structural error in the catalog.
- G — ALL ENTRIES EXCLUDED OR DUPS** — all catalog entries selected by this filter were discarded because:
  - 1) they were excluded by a preceding EXCLUDE statement
  - 2) they were previously selected by a preceding SELECT CATDSN= statement
- J — INSUFFICIENT STORAGE** — the REGION size was not large enough. Increase the REGION (use REGION=0M if you can) and resubmit.

**Action:** Serious errors will cause a U0502 ABEND.

**FATS990 INSTALLED INNOVATION TRIAL WILL EXPIRE ON yy.dd**

**Reason:** Product extension has completed successfully.

Required JCL:

```
//EXTEND EXEC PGM=FATEXTND,PARM=xxxx
//STEPLIB DD DISP=SHR,DSN=fats.loadlib
//SYSLIB DD DISP=SHR,DSN=fats.loadlib
//SYSDIAG DD SYSOUT=*
```

The PARM=value will be supplied by INNOVATION.



## 40.5 FATS ACTION MESSAGES

**ACTION  
MESSAGES**

The "ACTION" column of the FATS detail report will contain one of the following messages:

**3480 FORMAT FILE SENSED**

The current file is a normal 3480 file.

**3480XF FORMAT FILE SENSED**

The current file is an IDRC compacted 3480 file.

**3490E FORMAT FILE SENSED**

The current file is a 3490E format file which may or may not be IDRC compacted.

**3590 FORMAT FILE SENSED**

The current file is a 3590 (Magstar) format file which may or may not be IDRC compacted.

**DATA COMPACTED BY xx%**

The current file was IDRC compacted by the indicated amount.

**TAPE INDICATE**

The logical end of tape (end-of-tape reflector or cartridge logical end-of-tape) has been detected by a FATS WRITE operation.

**PERM DATA CHECK**

An error has been retried until the retry level defining a permanent error (RETRY= parameter) has been reached.

**PERM DATA CHECK CNTL**

An error was encountered while repositioning the tape for data check retry. No further retries are done. This may indicate serious problems with the tape or drive.

**TEMP DATA CHECK**

An error has been retried and was successful before the permanent error retry level was reached.

**TAPE MARK**

A tape mark (end of file) was read during a FATS read operation.

**LABEL SAVED**

The volume serial was read from the tape and labels were successfully rewritten (the SAVLAB function).

**LABEL NOT SAVED**

The SAVLAB function was attempted, but labels did not exist or could not be read.

**LABEL WRITTEN**

Labels with the specified volume serial number have been written to the tape.

**LABEL NOT WRITTEN**

Labels could not be written due to I/O errors.

**TEMP DATA CHK LABEL**

Temporary I/O errors were encountered when writing labels but the labels were written successfully.

**PERM DATA CHK LABEL**

Permanent I/O errors were encountered when writing or saving labels. The labels were not written.

**MAX ERROR EXCEEDED**

The length in inches of a contiguous data check has exceeded the MAXCERR= parameter specified. Processing of this tape will be terminated.

## 40.5 CONTINUED . . .

**ACTION  
MESSAGES  
(Continued)****PRINT THRESHOLD MET**

The number of errors printed for this tape exceeds the THRESHOLD= parameter specified. No more errors will be printed in the detail report, but processing will continue.

**OPERATOR CANCELLED**

Processing of this tape was cancelled by the operator by a reply of "KEOJ" or "Kuuu" to the FATSW02 console message or "TERM" or "SKIP" to the FATSW08 console message.

**OPEN ERROR OR NO DD**

Errors occurred opening this tape. Processing of all tapes on this tape unit will be halted.

**CLEAN ACTION REQUEST**

The number of errors on this tape exceeds the ERRCLEAN= parameter. The operator is requested to clean the tape drive and remount the same tape.

**90.0 FATS AND FATAR INSTALLATION****DISTRIBUTION  
TAPE FORMAT**

FATS and FATAR are distributed on standard label (SL) magnetic tape volume. The volume serial of the distribution tape will be:

FAT48P – a production (licensed) version of FATS alone or FATS/FATAR.

FAT48T – a trial (evaluation) version of FATS/FATAR (always contains both products).

The volume serial and status will be clearly marked on the external tape label. The trial programs will expire (i.e., stop functioning) on the expiration date indicated on the external label.

The files on the tape are:

- File 1: DSN=FAT.INSTALL  
an executable program which is used to load the files on the tape to disk.
- File 2: DSN=LOAD  
IEBCOPY unloaded copy of programs.
- File 3: DSN=DOC  
dummy file, no longer used.
- File 4: DSN=ICL  
IEBCOPY unloaded copy of installation jobstreams and supplementary documentation.
- File 5: DSN=CLIST  
IEBUPDTE-format file of ISPF dialog CLISTs.
- File 6: DSN=PANELS  
IEBCOPY unloaded copy of dialog PANELs.
- File 7: DSN=MESSAGES  
IEBCOPY unloaded copy of dialog MESSAGEs.
- File 8: DSN=SKELETON  
IEBCOPY unloaded copy of dialog SKELETONs.
- File 9: DSN=TABLES  
IEBCOPY unloaded copy of dialog TABLEs.
- File 10: DSN=JCL  
IEBCOPY unloaded copy of all example JCL from this user manual.

**90.1 LOADING THE FATS/FATAR LIBRARIES FROM THE DISTRIBUTION TAPE**

The FATS/FATAR Tape Install Program, FATLOAD, makes the installation of an FATS/FATAR distribution tape very easy. You can execute FATLOAD directly from tape if you have access to a TSO userid that has the MOUNT attribute, or if you are able to issue or request a command on a system console to have a tape mounted. Otherwise, you must copy the Tape Install Program to disk using the JCL shown in Step 3.

If you have access to a TSO userid with the MOUNT attribute, logon to that id and proceed to Step 2. If you don't know if your userid has the MOUNT attribute, you probably don't, so proceed to Step 1.

**Step 1** Use this step if your TSO userid does not have MOUNT privileges and you are able to issue (or request to be issued) a command on a MVS system console to have a tape mounted. If your userid has MOUNT privileges, proceed to Step 2; otherwise proceed to Step 3.

If your TSO userid does not have the MOUNT attribute, you can still access a tape from TSO by having the operator issue a MOUNT command. You or the operator must mount and ready the tape on a free tape drive **BEFORE** issuing the following command on a MVS system console:

**MOUNT uuu,VOL=(SL,FAT48T)** Change "uuu" to the actual tape unit address.

Change FAT48T to FAT48P if this is a production tape.

Now go to Step 2, but remember that when you are done with the tape, it must be unloaded by the MVS system command:

**UNLOAD uuu**

**NOTE:** if the tape unit has a 4-digit address, you must precede it with a slash, e.g.,

MOUNT /1234,VOL=(SL,FAT48T) and UNLOAD /1234

**Step 2** Use this step if your TSO userid has the MOUNT attribute, or if you have completed Step 1.

If you are using ISPF, issue the following TSO commands from ISPF Option 6 (TSO COMMANDS). You can also exit ISPF and issue them from the TSO "READY" prompt.

Enter this TSO command to allocate the FATS/FATAR distribution tape:

**ALLOC DA('FAT.INSTALL') VOL(FAT48T) UNIT(TAPE) SHR** Change FAT48T to FAT48P if you are loading from a production tape.  
Change the UNIT parameter to the appropriate tape unit name.

If you get the message "IKJ56221I DATASET FAT.INSTALL NOT ALLOCATED, VOLUME NOT AVAILABLE", it may be because your userid does not have the MOUNT attribute; go back to Step 1. If you have already done Step 1, then the problem is that the tape was mounted **AFTER** the MOUNT command was issued. Issue an UNLOAD console command and go back to step 1.

Now issue this TSO command to invoke the Tape Install Program:

**LOADGO 'FAT.INSTALL'**

The Tape Install program (FATLOAD) will be loaded from the tape and begin execution. Proceed to Step 4.

## 90.1 CONTINUED . . .

- Step 3** Use this step to submit a batch job to copy the Tape Install Program to a disk file, from which it can be executed under TSO.

Submit this jobstream:

```
//IEBGENER EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DISP=(,CATLG),SPACE=(3200,10),
//          DSN=user-specified-name. <=== specify a dataset name
//          UNIT=SYSALLDA,
//          VOL=SER=vvvvvv <=== specify a disk volume
//SYSUT1 DD DISP=OLD,DSN=FAT.INSTALL,
//          LABEL=(1,EXPDT=98000),
//          UNIT=TAPE, <=== change if required
//          VOL=SER=FAT48T <=== change to FAT48P if production tape
```

After the successful completion of the IEBGENER job, issue this TSO command from ISPF Option 6 (TSO COMMANDS) or the TSO READY prompt:

**LOADGO 'user-specified-name'** Specify the same dataset name given in the JCL, in quotes.

The Tape Install program will be loaded from disk and begin execution. Proceed to Step 4.

- Step 4** The tape install program will prompt you for information on what, where and how to load the FATS/FATAR tape files, in a series of four user-friendly screens. **No action will take place until you give the final confirmation on the fourth screen.** Only then are the output datasets allocated and cataloged with the names you specified, and the loading of those datasets begins (either in the foreground or via a batch jobstream).

**NOTE:** all dataset names and index name references are specified and displayed as fully-qualified names: a TSO userid will **not** be prefixed to the names unless you key it in.

**Screen 1 -  
DataSet  
Selection**

```
WELCOME TO INNOVATION DATA PROCESSING FATS/FATAR INSTALLATION      SCREEN 1

PLEASE REPLY TO THE FOLLOWING PROMPTS. YOU WILL BE ABLE TO REVIEW AND
CHANGE YOUR SPECIFICATIONS PRIOR TO THE ACTUAL LOADING OF THE TAPE.

THE FOLLOWING DATASETS MAY BE LOADED FROM THE INSTALLATION TAPE:

  1 - FATS/FATAR INSTALLATION CONTROL LIBRARY
  2 - FATS/FATAR LOAD MODULE LIBRARY
  3 - FATS/FATAR ISPF DIALOG CLIST LIBRARY
  4 - FATS/FATAR ISPF DIALOG PANEL LIBRARY
  5 - FATS/FATAR ISPF DIALOG MESSAGES LIBRARY
  6 - FATS/FATAR ISPF DIALOG SKELETON LIBRARY
  7 - FATS/FATAR ISPF DIALOG TABLE LIBRARY
  8 - FATS/FATAR JCL LIBRARY

-----
<PRESS>  "ENTER"      -  SELECT ALL OF THE ABOVE DATASETS AND CONTINUE
<TYPE>   "N,N;.."     -  SELECT THE SPECIFIED DATASETS
<TYPE>   "END"        -  EXIT IMMEDIATELY
-----
PLEASE SELECT ONE OF THE OPTIONS LISTED ABOVE
SELECT ==>
```

This screen allows you to select which of the datasets are to be loaded from the FATS/FATAR distribution tape. Normally, all datasets should be selected. When you are satisfied with the selection, press ENTER to continue to Screen 2.

## 90.1 CONTINUED . . .

Screen 2 -  
DataSet Name  
Selection

```

----- DATASET NAME SELECTION SCREEN ----- SCREEN 2
PLEASE REVIEW THE SELECTED DATASET NAMES AND MAKE THE DESIRED MODIFICATIONS.

1 - INSTALL CONTROL..... FATS.ICL48
2 - LOAD LIBRARY..... FATS.LOAD48
3 - ISPF CLISTS..... FATS.CLIST48
4 - ISPF PANELS..... FATS.PANELS48
5 - ISPF MESSAGES..... FATS.MSGS48
6 - ISPF SKELETON..... FATS.SKELS48
7 - ISPF TABLES..... FATS.TABLES48
8 - JCL..... FATS.JCL48

-----
<PRESS>  "ENTER" - USE THE ABOVE SPECIFICATIONS AND CONTINUE
<TYPE>   "ALL,INDEX" - ASSIGN NEW INDEX(ES) TO ALL DATASET NAMES
<TYPE>   "N,NEWNAME" - ASSIGN A NEWNAME TO THE DATASET DESIGNATED BY "N"
<TYPE>   "BACK" - GO BACK TO THE DATASET SELECTION SCREEN 1
<TYPE>   "END" - EXIT IMMEDIATELY
-----
PLEASE SELECT ONE OF THE OPTIONS LISTED ABOVE
SELECT ==>

```

This screen allows you to specify the dataset names which will be used for the datasets you have selected to load from the tape. These may be existing datasets to be updated, or they may be new datasets which will be allocated and cataloged.

The names shown above are the default names provided with the Tape Install program. You may change these names in one of 2 ways:

1) to change the current high-level index of all of the datasets to a different index (or indexes), enter "ALL,newindex(s)". For example,

ALL,FAT48 will change the names to FAT47.ICL48, etc.

ALL,SYS3.FAT48 will change the names to SYS3.FAT47.ICL48, etc.

2) to completely change the name of any one dataset, enter that dataset's number followed by the replacement name. For example,

2,SYS2.FATS.LOAD will change the name of the load library.

You can use either or both of these techniques repeatedly until you are satisfied with the names.

If you intend to update an existing library, be sure that library name is correctly specified. However, we recommend that you always install into newly created libraries to avoid X37 abends due to insufficient space in existing libraries. For new datasets, the install program will allocate them with sufficient space.

The FATS/FATAR load library must be defined to MVS as an authorized library. If it is not already defined as authorized, you (or your MVS system programmer) must do so by:

- For MVS releases prior to ESA V4, update the IEAAPFxx member in PARMLIB. An IPL will be required to activate the updated authorized library list.
- For ESA V4 and V5 and OS/390, if your installation has converted to the use of the PROGxx member in PARMLIB to define authorized libraries, you can update the PROGxx member and activate the updated list with the MVS console command:

**SET PROG=xx**

If your installation has specified that the authorized program list is in dynamic format, you can also temporarily authorize the FATS/FATAR library with the MVS console command:

**SETPROG APF,ADD,DSNAME=fats.load.library,VOL=volser**

- You may also have another means of temporarily authorizing the FATS/FATAR load library for testing. Remember that temporary authorization will vanish after the next IPL unless the appropriate PARMLIB member is updated.

## 90.1 CONTINUED . . .

Screen 3 -  
Volume Serial  
Selection

```

----- VOLUME SERIAL SELECTION SCREEN ----- SCREEN 3
THE FOLLOWING NEW DATASETS WILL BE ALLOCATED AND CATALOGED:

      DISP  VOLUME  DATASET NAME
1 - INSTALL CONTROL NEW      FATS.ICL48
2 - LOAD LIBRARY... NEW      FATS.LOAD48
3 - ISPF CLISTS... NEW      FATS.CLIST48
4 - ISPF PANELS... NEW      FATS.PANELS48
5 - ISPF MESSAGES... NEW     FATS.MSGS48
6 - ISPF SKELETON... NEW     FATS.SKELS48
7 - ISPF TABLES... NEW     FATS.TABLES48
8 - JCL... NEW              FATS.JCL48

-----
<PRESS>  "ENTER" - USE THE ABOVE SPECIFICATIONS AND CONTINUE
<TYPE>   "ALL,VOLUME" - ASSIGN A VOLUME TO ALL NEWLY ALLOCATED DATASETS
<TYPE>   "N,VOLUME" - ASSIGN A VOLUME TO THE DATASET DESIGNATED BY "N"
<TYPE>   "SMS" - DISPLAY SMS SPECIFICATIONS
<TYPE>   "BACK" - GO BACK TO THE DATASET NAME SELECTION SCREEN 2
<TYPE>   "END" - EXIT IMMEDIATELY

-----
PLEASE SELECT ONE OF THE OPTIONS LISTED ABOVE
SELECT ==>

```

On this screen, the DISP column shows whether the install program found that the indicated dataset already exists (OLD) or does not exist (NEW). For NEW datasets, you can specify volume and/or SMS information to be used for the allocation of the datasets. For OLD datasets, the VOLUME column shows the volume serial of the existing dataset.

Similar to Screen 2, you can specify the target disk volume serial for all or any one of the datasets to be allocated. For example,

ALL,SYSLB2 will change the target volume serial for all the datasets.

3,BOOKM2 will change the target volume serial for the CLIST library.

You can use either or both of these techniques repeatedly until you are satisfied with the names. The volume serial can be omitted if the dataset will be SMS-managed or if your system will allocate such datasets on non-specific storage volumes.

If SMS is active on your system, you may enter "SMS" which will take you to variations of Screen 3 which will allow you to specify the SMS storage class, management class, and/or data class to be assigned to each dataset. However, this is not necessary if your installation's SMS ACS routines will assign proper classes to these datasets.

## 90.1 CONTINUED . . .

Screen 4 -  
Installation  
Processing  
Option

```

----- INSTALLATION PROCESSING OPTION SCREEN ----- SCREEN 4
PLEASE VERIFY THE FOLLOWING SPECIFICATIONS AND SELECT THE PROCESSING OPTION:

      DISP  VOLUME  DATASET NAME
1 - INSTALL CONTROL NEW          FATS.ICL48
2 - LOAD LIBRARY.... NEW          FATS.LOAD48
3 - ISPF CLISTS.... NEW          FATS.CLIST48
4 - ISPF PANELS.... NEW          FATS.PANELS48
5 - ISPF MESSAGES... NEW          FATS.MSGS48
6 - ISPF SKELETON... NEW          FATS.SKELS48
7 - ISPF TABLES.... NEW          FATS.TABLES48
8 - JCL..... NEW          FATS.JCL48
-----
<TYPE>  "FG"      - START LOADING THE ABOVE DATASETS IMMEDIATELY
<TYPE>  "BG"      - CREATE THE JCL TO LOAD THE ABOVE DATASETS
<TYPE>  "BACK"    - GO BACK TO THE DATASET SELECTION SCREEN 1
<TYPE>  "END"     - EXIT IMMEDIATELY
-----
PLEASE SELECT ONE OF THE OPTIONS LISTED ABOVE
SELECT ==>

```

On this screen, you can review all of the decisions you have made before starting the actual loading of the libraries from the distribution tape. Entering "BACK" on this screen (or any of the others) will allow you to go back and change options before installation.

If **FG** (foreground) installation is chosen, all the datasets indicated as NEW will be allocated, then IEBCOPY or other utilities are invoked under TSO to load each of the selected libraries from tape. This option appears only if you loaded the Tape Install program directly from tape (Step 2). FG is recommended since the tape is already mounted.

If **BG** (background) installation is chosen, the NEW datasets will be allocated under TSO (same as FG) but then batch JCL will be created to actually load the libraries from the tape. This JCL will be stored as member FATLOAD in the ICL (Installation Control Library) you specified, or, if you did not select the ICL, in a dataset named "userid.FATSTEMP.JCL". You must review this jobstream, make any changes necessary for your installation, and submit it for execution.



## 90.2 ALTERNATE MANUAL INSTALLATION

If you prefer not to use the recommended interactive install program, you can load the libraries to disk manually.

First, copy the install jobstream from the ICL (Installation Control Library) on tape to disk, using JCL similar to:

```

JCL FOR      //INSTALL      JOB      ...
LOADING     //COPY        EXEC    PGM=IEBCOPY
INSTALL     //SYSPRINT    DD      SYSOUT=*
JOBSTREAM  //TAPEIN      DD      DSN=ICL,DISP=OLD,LABEL=4,UNIT=tape,
//              VOL=SER=FAT48T    << change to FAT48P if production tape
//              DD          DSN=your.library.here,
//              DISP=SHR
//SYSIN       DD      *
COPY INDD=((TAPEIN,R)),OUTDD=LIBRARY
SELECT MEMBER=INSTALL
//*
```

You will need to specify an appropriate unit name for the tape; if your tape is a production tape, be sure and change the tape volume serial to FAT48P. The LIBRARY DD statement should point to a control card library (RECFM=FB and LRECL=80); this can be an existing library or you can allocate a new one. The jobstream above will load **only** the single member **INSTALL**.

If you want to **only** install the FATS/FATAR Load Library use the member **INSTMIN**.

**INSTALLATION**  
**JCL**

The JCL member loaded by the preceding job provides you with procedures for the installation of the FATS/FATAR:

- Product load library.
- Installation Control library
- JCL (example) library
- ISPF dialog libraries

The JCL stream consists of three in-stream procedures and the necessary steps to execute those procedures to install the FATS/FATAR libraries. One procedure allocates the target libraries and contain the recommended size of each library; the other two execute IEBCOPY or IEBUPDTE to load the library from tape to disk. The jobstream contains comments which will guide you through modifying the jobstream to substitute dataset names and target disk volumes of your choice. Once modifications are complete, submit the jobstream to allocate and load the libraries.

**NOTE:** The FATS/FATAR load library must be an authorized program library as shown in Section 90.1.

**90.3 IMPLEMENTING THE TAPE MANAGEMENT INTERFACE**

An interface to tape management systems is included to get complete information about the volumes and datasets being read by FATAR. It is also used by FDRCOPY when selecting and copying datasets and is used by the FATS/FATAR ISPF dialogs to query tapes to be processed. Details on this interface are in Section 21.5.

DFSMSrmm (RMM) from IBM and CA-1 (TMS) and CA-DYNAM/TLMS, both from Computer Associates, are supported.

The support for DFSMSrmm requires that the RMM API (Application Program Interface) is available. If you are running DFSMS 1.3 you must have IBM PTF UW47006 installed. DFSMS 1.4 users must have PTF UW47007 installed. Both of these PTFs were available in 1998. DFSMS 1.5 and above includes the API, which means that OS/390 2.7 and above, and all releases of z/OS, can support our RMM interface.

**ASSEMBLING  
THE INTERFACE**

To implement this interface you must assemble a piece of the interface code using the macro library supplied with your current tape management system. These macros include the necessary code to retrieve information from the tape management data base and the layouts of the tape management records.

Members in the FATS/FATAR ICL (Installation Control Library) contain the assembly jobstreams and instructions for doing the assembly. Depending on the type of your tape management system, please edit and submit one of the following ICL members, following the instructions in the member:

ASMCA1    – for CA-1 (TMS)  
ASMTLMS   – for CA-DYNAM/TLMS  
ASMRMM    – for DFSMSrmm

If you previously assembled the tape management interface for use in a previous release of FATS, you should reassemble it to get the latest code updates.

**NEW RELEASES  
OF TAPE  
MANAGEMENT**

The assembly procedure will assemble the interface module using the macros provided with the current version of your tape management system. If you install a new version of your tape management system, you should reassemble the FATS/FATAR interface so that any changes in the structure of the tape management data or interfaces will be automatically accommodated. However, don't make the reassembled interface generally available until you install the new tape management system in production.

**TESTING THE  
INTERFACE**

You can quickly test the tape management interface using a test program, FATTMTST. The JCL is simple:

```
//TESTTMS EXEC PGM=FATTMTST,PARM=volser
//STEPLIB DD DISP=SHR,DSN=fatsfatr.load.library
//SYSPRINT DD SYSOUT=*
```

The parameter (PARM=volser) is a tape volume serial. FATTMTST will use the tape management interface to query that volume and will print out the dataset name on the volume. If the tape has multiple files on it, all tape files are listed. If the tape is part of a multi-volume tape set, all volumes in the set (and all datasets on the set) are listed. If the volume serial is not recorded in your tape management data base, you will get a return code of 12. If you get other return codes, abends, or unexpected results, please contact Innovation for assistance.

## 90.4 CUSTOMIZING FATS AND FATAR

At this time there are no customizable options or settings in FATS and FATAR. All options are set via control statement operands.

**OPTIONAL  
MODULES AND  
TABLES**

However, the ICL (Installation Control Library) contains three source module which you may wish to modify and assemble.

Member **OPENEXIT** is a sample FATAR Open exit. Use of the Open exit is documented in Section 21.4 and is controlled by operand EXIT= documented in Section 23.2.5.

Member **FATCHTAB** is a sample FATAR printer translate table. Use of the translate table is documented under the CHARTAB= operand in Section 23.2.4.

Member **ASCIITRS** is a sample FATAR ASCII translate table, containing a table for translation of ASCII to EBCDIC and another for EBCDIC to ASCII. Use of the translate table is documented under the ASCII= operand in Section 23.2.4.

**SECURITY**

By default, FATAR will do security checks on input datasets. It will verify that users have READ authority to each input dataset in the DATASET class. This prevents users from using FATAR to access data to which they are not authorized, especially on multi-file tapes. More details are found in Section 21.6. If you do not want to do this dataset security check, you have two options:

- 1) you can apply FATAR custom zap C-48.0001 to disable the security check, or
- 2) you can define security resource FATAR.DATASET.SECBYPAS in class FACILITY and give all users READ authority to it.

By default, all users can use all functions and options of FATS and FATAR. If you want to restrict certain users from using certain functions, you must apply FATS/FATAR custom zap C-48.0001 and define FACILITY class resources in your security system to control each function, as described in Sections 11.8 (FATS security) and 21.6 (FATAR security).

**90.5 INSTALLING THE ISPF DIALOG**

The FATS/FATAR ISPF dialog can be invoked at any time, from any TSO userid that is authorized to read the FATS/FATAR dialog libraries, by going to ISPF option 6 (TSO COMMAND) and entering:

**EXEC 'fats.clist.library(FATALLOC)'**

However, this is awkward, so two shortcuts are available:

**ADDING FATS/  
FATAR TO A  
MENU**

You may add FATS/FATAR as an option on the ISPF main menu or any menu of your choice. In the FATS/FATAR panel dataset, there are two example panels showing how to add FATS/FATAR to the ISPF main menu:

**ISR@V3X** – for ISPF V3.x and V2.x

**ISR@V4X** – for ISPF V4.x

You can make similar modifications to your ISPF main menu (panel ISR@PRIM) or to any other ISPF menu.

**ADDING FATS/  
FATAR AS A  
COMMAND**

You can add a FATS/FATAR command to the ISPF command table (ISPCMDS) by going to ISPF option 6 (TSO COMMAND) and entering:

**EXEC 'fats.clist.library(FATCMDS)'**

The modified command table will be stored in the first library in the ISPTLIB concatenation for this TSO userid. If the user has a private table library, it will normally be the first one in that concatenation, so the updated command table will be available only to this user. If the user does not have a private library, but has update authority to the first public library in ISPTLIB, it will be updated, and will be available to any userid using that library. However, you will have an option to specify a different table library before it is actually stored. If you do choose a different table library, that library will have to be in the ISPTLIB concatenation, prior to any other library containing a table called ISPCMDS. The FATS/FATAR dialog can now be entered from almost any ISPF panel by entering "FATS" on the command line.

**CHANGING  
THE DIALOG  
LIBRARY  
NAMES**

During the installation with the FATLOAD program, the names of the FATS/FATAR dialog libraries are automatically changed in the CLIST library to the names you specified on the installation screens. If you later change those names, or if you manually loaded the dialog libraries, you can correct the dialog to use the proper names by editing member FATALLOC in the FATS/FATAR CLIST library. Change the appropriate CLIST lines to specify the new dataset name(s):

CLIST FATS/FATAR ISPF Dialog CLIST Library

LLIB FATS/FATAR Load Module Library

PLIB FATS/FATAR ISPF Dialog Panel Library

MLIB FATS/FATAR ISPF Dialog Message Library

SLIB FATS/FATAR ISPF Dialog Skeleton Library

TLIB FATS/FATAR ISPF Dialog Table Library

Do not modify the FATSFATR CLIST for this purpose. The table library must be allocated before FATSFATR is invoked.

**90.6 VIEWING/PRINTING THE SOFTCOPY DOCUMENTATION****VIEWING/  
PRINTING THE  
SOFTCOPY  
DOCUMENTA-  
TION**

A copy of this FATS/FATAR manual is provided on a CD-ROM in softcopy format for use with IBM BOOKMANAGER products.

The CD-ROM contains a copy of the IBM Softcopy Reader, which enables you to read the Bookmanager documents.

In addition, the CD-ROM contains the manual in Adobe Acrobat PDF format, plus a copy of the Adobe Acrobat Reader, enabling you to view and print the manual in a format identical to the printed manual.

The manuals on the CD-ROM can be used directly on any of the supported systems except MVS. To use Bookmanager files on MVS, you must have READ/MVS and you must upload the Bookmanager files to MVS disk storage, following directions on the CD-ROM.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

**93.0 ISPF INTERFACE INTRODUCTION**

FATS/FATAR ISPF panels are available to perform many of the FATS and FATAR functions like labeling, certifying and copying tapes. These panels create FATS or FATAR JCL and control statements as jobstreams which can be submitted, edited before submission, or saved for reuse.

The panels will simplify the use of FATS/FATAR for common tasks. Extensive validation of the input is done to eliminate errors at execution time. More complicated tasks such as dropping blocks or replacing data on a tape will be easier to perform.

Options for installing and invoking the FATS/FATAR dialogs are shown in Section 90.4.

Every panel has detailed HELP information which can be displayed by pressing the HELP key (usually PF1). If your level of ISPF supports it, field level help is available by placing the cursor in the input field in question and pressing HELP. The descriptions of the fields and functions on the panels in the following sections are brief because the detailed, current descriptions are in the HELP.

The actual ISPF panels may vary slightly from the samples in this manual.

**MAIN MENU**

----- FATS/FATAR MAIN MENU -----			
OPTION	==>		
A. CERTIFY	new tapes		
B. CERTIFY	existing labeled tapes whose datasets have expired		
C. LABEL	new tapes		
E. ERASE	old data on tapes for security		
F. VERIFY	whether tapes are still readable		(FATS READ)
G. VERIFY	whether tapes are still readable		(FATS ANALYZE)
H. EXAMINE	contents and layout (map) of foreign or unknown tapes		
J. COPY	a tape volume - (make exact image)		
K. COPY	multiple volumes - (make exact image)		(FATS ANALYZE)
L. COPY	one or more files - (make a logical copy)		
M. FATAR	extended functions		
O. RECOVER	data from overwritten (damaged) tape		
Q. QUERY	Tape Management System - CA-1/TMS (V5.1)		
S. SETUP	JCL and FATS/FATAR defaults for user		(Profiles)

This is the FATS/FATAR Main Menu. On your first use of this dialog, you should select the SETUP (S) option first, to set up the Job Profile and the FATS/FATAR defaults for your TSO USERID. Once that has been done, you may select other options from the menu to generate and submit jobstreams for common FATS and FATAR options. Note that options G through O will work only if you are licensed for FATAR.

**TAPE  
MANAGEMENT  
SUPPORT**

If you have installed the FATS/FATAR Tape Management Interface (see Section 90.3), option **Q** will appear on this menu. As shown above, it identifies the type and version of tape management which FATAR has identified as being active on your system. Option Q allows you to query your tape management data base through facilities provided by FATS/FATAR.

For a given tape volume serial, option Q will display all the datasets recorded as being on that tape. If the tape is part of a multi-volume tape set, all volumes and all datasets in the set are displayed, also showing which volumes each dataset resides on.

Also, when the tape management interface is installed, you may also use a Q on the COMMAND line of many of the panels for the various FATS and FATAR functions (a prompt will appear on the panel when the function is available). This allows you to query your tape management system and automatically copy dataset name, label type, and file number information into the appropriate fields for the input or output tapes.

## 93.0 CONTINUED . . .

**FATS/FATAR  
SETUP/PROFILE  
PANELS**

The following series of panels is invoked by option S (SETUP) on the FATS/FATAR main menu. They set defaults that will be used during the generation of jobstreams by other options on the menu. There are 3 kinds of defaults:

- JCL options, for generation of JOB statement and other JCL parameters
- FATS and FATAR “Profile” options, giving values for operands, such as lines/page, which **cannot** be overridden on other panels. The defaults provided by Innovation for these values will rarely need to be overridden, but if necessary you can set them to the values you need here.
- FATS and FATAR “Control” options, giving values for operands which **can** be overridden on other panels. However, setting these to the values you most commonly use will save time and typing later.

You can return to these panels at any time to change your saved options. These options are saved in your ISPF PROFILE dataset, so they affect only your TSO userid.

On the FATS and FATAR function panels, an option will always appear asking if you want to see more FATS or FATAR operands. If you reply YES, you will see a panel similar to the “Control” panels shown below. If you change values on those panels, they are **not** saved in your profile and will need to be reentered every time you need them.

**JOB PROFILE**

```

----- FATS/FATAR JCL PROFILE -----
COMMAND ==>

JOB STATEMENT:
==> //userid      JOB (ACCOUNT),'NAME'
==> // *
==> // *
==> // *

LOAD LIBRARIES:
==> ----- <
==> ----- <

SYSOUT CLASS:
==> *

VOLSER DEFAULT:
==> 99999 (default for Input DD Statement; 6th digit is system generated)

ADD Job Id Character: YES (yes/no)    SAVE DSN changes in Profile: NO (yes/no)

Press DOWN for FATS Profile

```

This panel allows you to enter any JOB statement parameters required by your installation and other defaults to be used when generating JCL for FATS/FATAR. If FATS/FATAR is in the system link list, no LOAD LIBRARY is required; if not, provide the name of the FATS/FATAR load library and STEPLIB DD statements will be generated. When done, press the DOWN key (usually PF8 or PF20) to see the next panel or the END key (usually PF3 or PF15) to save the profile.



## 93.0 CONTINUED . . .

**FATS DEFAULT  
PROFILE  
OPERANDS**

```

----- FATS DEFAULT "PROFILE" OPERANDS -----
COMMAND ==>
MODIFY   - Respond to operator's MODIFY(F) and STOP(P) commands : YES (yes/no)
WTOR     - Issue WTOR message for operator control of processing: NO  (yes/no)
MAXCLEAN - Clean tape drive after --- tapes are processed on drive
LINECNT  - Number of lines per page to be printed on FATS reports : 56
MAXDETCNT - Number of detail lines to be printed before double spacing: 4
NEWPAGE  - Skip to new page when a new volume is processed : NO  (yes/no)
METRIC   - List tape lengths/error positions in metric units: NO  (yes/no)
RETCODE  - Terminate with return code 12 due to major error : YES (yes/no)
WTO      - Report result of each operation on system console: NO  (yes/no)
REWIND   - Rewind tapes to load point at start of operation : YES (yes/no)

Press DOWN for FATS Control Operands          Press UP for JOB Profile

```

This panel modifies the default options that will be used for many FATS operations. These operands are **not** overridable from the function panels. Most users will require no changes except for those users who would prefer the lengths be shown in metric (the METRIC option). When done, press the DOWN key (usually PF8 or PF20) to see the next panel or the END key (usually PF3 or PF15) to save the profile.

**FATS DEFAULT  
CONTROL  
OPERANDS**

```

----- FATS DEFAULT "CONTROL" OPERANDS -----
COMMAND ==>
BLKSIZE  - Size of block written to new tape during Certification : 20---
NUMWRITES - Nbr of writes that FATS will do in one I/O when Certifying: 20---
RETRY(H) - Number of retries before error is considered permanent : 10
RETRY(I) - Number of retries before error is printed in detail report: 1
ERRCLEAN - Clean tape drive after 32767 temporary or permanent errors
MAXERR    - Stop processing after 200 permanent errors encountered
MAXCERR   - Stop processing after 200 inches of contiguous permanent errors
MAXTERR   - Stop processing after 1000 temporary data checks are detected
THRESHOLD - Stop printing error messages after 1000 permanent/temporary errors
STOPNUM   - Invoke STOP option after permanent errors have been encountered
STOP      - Stop/Issue FATSW05 message for each permanent error: NO  (yes/no)
DDLABEL   - Label tape with volume serial in TAPE1 DD statement: NO  (yes/no)
LABEL     - Label tape with volume serial nbr: -----
MODE      - Recording Mode for Drive : ----- (refer to tutorial/manual)
OPERATOR  - Invoke OPERATOR/NOOPERATOR option: ---- (oper/noop)
OWNERID   - Contents of owner ID field in output volume label: -----

Press DOWN for FATAR Operands          Press UP for FATS Operands

```

This panel modifies the default options that will be used for many FATS operations. These operands **are** overridable from the function panels. A subset of this panel is displayed whenever you answer YES to the question "Display More FATS Operands?" on the function panels. When done, press the DOWN key (usually PF8 or PF20) to see the next panel or the END key (usually PF3 or PF15) to save the profile.

**FATAR  
DEFAULT  
PROFILE  
OPERANDS**

```

----- FATAR DEFAULT "PROFILE" OPERANDS -----
COMMAND ==>
MAJERR    - Stop processing after 01 major errors are encountered
MAXERR    - Stop processing after 20 permanent data checks are detected
MAXTERR   - Stop processing after 100 temporary data checks are detected
LINECNT   - Number of lines per page to be printed on FATAR reports: 60
BYTEFACTOR - Specify divisor to calculate byte count values for reports: 1000
METRIC    - List tape lengths/error positions in metric units: NO  (yes/no)
RETCODE   - Terminate with return code 12 due to major error : YES (yes/no)
WTO       - Report result of each operation on system console: NO  (yes/no)
ROR       - Attempt "Read-Opposite-Recovery" on round reels : YES (yes/no)
MSGLEVEL  - Print contents of modified blocks? : YES (yes/no)
EXITNAME  - Name of load module to be executed at open time : -----

Press DOWN for FATAR Control Operands          Press UP for FATS Operands

```

This panel modifies the default options that will be used for many FATAR operations. These operands are **not** overridable from the function panels. Most users will require no changes except for those users who would prefer the lengths be shown in metric (the METRIC option). When done, press the DOWN key (usually PF8 or PF20) to see the next panel or the END key (usually PF3 or PF15) to save the profile.

## 93.0 CONTINUED . . .

**FATAR  
DEFAULT  
CONTROL  
OPERANDS**

```

----- FATAR DEFAULT "CONTROL" OPERANDS -----
COMMAND ==>
RETRY      - Number of retries before error will be considered permanent: 40
PRTLEN     - Number of bytes/characters of input tape data to be printed: 80
DATAFMT    - Print format of blocks/records: DUMP (char/hex/dump)
LBLPRT     - Print format of input labels : FORMAT (char/hex/dump/format/none)
LENCHK     - Print contents of different block lengths: NO (yes/no)
STOP       - Provide operator control at End-of-File : NO (yes/no)
PRINTTMS   - Print TMS list for selected Volume Serial: NO (yes/no)
KEEP       - Copy block even on a permanent data check: NO (yes/no)
EOV        - Check for EOVS labels/update block counts : YES (yes/no)
EOTBLOCKS  - Number of blocks to write after EOT for Image copy: --
CREATEDATE - Copy creation date from input tape labels: NO (yes/no)
USERLABELS - Copy user labels to output tape labels : YES (yes/no)
MAXRBLK    - Largest current blocksize for reblocking : -----
REBLOCK    - Target blocksize for data reblocking : -----
ASCII      - Name of load module containing translation table : NO
CHARTAB     - Name of load module containing translation table : -----

```

Press UP for FATAR Operands

This panel modifies the default options that will be used for many FATAR operations. These operands **are** overridable from the function panels. A subset of this panel is displayed whenever you answer YES to the question "Display More FATAR Operands?" on the function panels. When done, the END key (usually PF3 or PF15) to save the profile.

## 93.1 FATS ISPF INTERFACE

**FATS  
OPERATIONS**

Options A through G on the FATS/FATAR main menu will generate jobstreams for FATS operations (option G will work only if you are also licensed for FATAR).

Once you enter all necessary options on a given panel, press ENTER to complete generation of the FATS batch jobstream. You will be given the option of submitting the jobstream immediately or entering ISPF EDIT to view and modify the jobstream; you can submit it directly from edit, or save it in a JCL library

All the panels contain a dataset name, unit name, and label parameter. A unique default dataset name may be generated but you may overwrite it with another name (this may be required to meet security restrictions); the name must be present to meet JCL requirements, but is usually not actually used by FATS. The unit name should be overridden with a generic name (such as 3490) or esoteric name (such as CART) which will allocate a tape drive of the appropriate type; a specific unit name (such as 3A0) may also be used. The label type presented by the panel is usually appropriate for the function requested but you may overwrite it if required.

All panels contain those FATS options which are commonly used with the function on that panel; you can override the defaults displayed as necessary. They also contain a question "Display more FATS Operands?". If you enter YES, the FATS DEFAULT CONTROL OPERANDS panel shown in Section 93.0 is displayed, where you can override less commonly used parameters. The overrides are effective for this function only; the values are not saved.

**FATS CERTIFY  
NEW TAPES**

```

----- FATS CERTIFY NEW TAPES -----
COMMAND ==>                                     Enter Q for TMS Query
TAPE1 DD Dsname= BAB.FATSOUT.JOBT                                     <
        Unit  =( TAPE )
        Label =( ,BLP,EXPDT=98000 )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                       5:  -----  6:  -----  7:  -----  8:  -----
                                       9:  ----- 10:  ----- 11:  ----- 12:  -----
                                      13:  ----- 14:  ----- 15:  ----- 16:  -----
                                      17:  ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :  NO-
NOLABEL - Create no label (NL) tapes:  NO- (yes/no)
ANSI     - Create ANSI labeled tapes :  NO  (yes/no)
VALIDATE - Verify Volser/Expired date:  NO  (no/yes)

Display more FATS Operands?: NO (yes/no)

```

Option A (CERTIFY new tapes) on the FATS/FATAR Main Menu brings you to this panel, to certify and label new tapes (or relabel existing tapes). Normally it creates IBM standard labeled tapes, but you can create ANSI (AL) tapes or unlabeled tapes by specifying YES to the proper option. Since these are usually virgin (never used) tapes, the BLP option must be used to mount them without trying to read existing labels.

Tape volume serials can be specified in two ways:

- specify up to 20 specific volume serials
- specify the first volume serial after "1:" and specify the number of volumes after "MAXVOLN". A range of sequentially number volumes will be initialized.

## 93.1 CONTINUED . . .

**FATS CERTIFY  
EXISTING  
TAPES**

```

----- FATS CERTIFY EXISTING TAPES -----
COMMAND ==>                                     Enter Q for TMS Query
TAPE1 DD Dsname= BAB.FATSOUT.JOBT                <
        Unit  =( TAPE )
        Label =( ,SL )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                     5:  -----  6:  -----  7:  -----  8:  -----
                                     9:  ----- 10:  ----- 11:  ----- 12:  -----
                                    13:  ----- 14:  ----- 15:  ----- 16:  -----
                                    17:  ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :  NO- (yes/no)
NOLABEL - Create no label (NL) tapes :  NO- (yes/no)
ANSI     - Create ANSI labeled tapes :  NO (yes/no)
VALIDATE- Verify Volser/Expired date: NO (no/yes)

Display more FATS Operands?: NO (yes/no)

```

Option B (CERTIFY existing labeled tapes) on the FATS/FATAR Main Menu brings you to this panel, to certify existing labeled tapes. This might be used to periodically recertify tapes while they are in scratch status. Since these are existing labeled tapes, SL allows MVS to verify that the proper tapes is mounted and in scratch status before beginning the certification.

If you have a range of tapes to recertify, you can specify a starting serial number and the maximum number of volumes. It is more likely that you will have a list of random serials to process, simply enter up to 20 such serials.

**FATS LABELING  
OF NEW TAPES**

```

----- FATS LABEL NEW TAPES -----
COMMAND ==>                                     Enter Q for TMS Query
TAPE1 DD Dsname= BAB.FATSOUT.JOBT                <
        Unit  =( TAPE )
        Label =( ,BLP,EXPDT=98000 )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                     5:  -----  6:  -----  7:  -----  8:  -----
                                     9:  ----- 10:  ----- 11:  ----- 12:  -----
                                    13:  ----- 14:  ----- 15:  ----- 16:  -----
                                    17:  ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :  NO- (yes/no)
NOLABEL - Create no label (NL) tapes :  NO- (yes/no)
ANSI     - Create ANSI labeled tapes :  NO (yes/no)
VALIDATE- Verify Volser/Expired date: NO (no/yes)

Display more FATS Operands?: NO (yes/no)

```

Option C (LABEL new tapes) on the FATS/FATAR Main Menu brings you to this panel, which labels (but does not certify) new tapes or relabels existing tapes. The parameters are the same as for Option A (Certify New Tapes).

## 93.1 CONTINUED . . .

**FATS ERASE  
TAPE**

```

----- FATS ERASE OLD DATA -----
COMMAND ==>                                     Enter Q for TMS Query

TAPE1 DD Dsname= BAB.FATSOUT.JOBT
        Unit  =( TAPE )
        Label =( ,SL )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                     5:  -----  6:  -----  7:  -----  8:  -----
                                     9:  ----- 10:  ----- 11:  ----- 12:  -----
                                    13:  ----- 14:  ----- 15:  ----- 16:  -----
                                    17:  ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :
SAVLAB  - Preserve volume serial?    : YES (yes/no)
NOLABEL - Create no label (NL) tapes: NO  (yes/no)
ANSI    - Create ANSI labeled tapes : NO  (yes/no)
VALIDATE- Verify Volser/Expired date: NO  (no/yes)

Display more FATS Operands?: NO  (yes/no)

```

Option E (ERASE) on the FATS/FATAR Main Menu brings you to this panel, which allows you to erase all data from one or more tapes. This might be used before selling old tape volumes or before writing a volume with data to be sent offsite, to ensure that all previous corporate data is removed. The parameters are the same as for Option B (Certify Existing Tapes), except for the SAVLAB option which allows you to preserve the existing volume serial.

**WARNING:** This option will erase all data on a tape volume. Use caution when using this option.

**FATS VERIFY OF  
EXISTING DATA  
ON TAPE**

```

----- FATS VERIFY (READ) TAPES -----
COMMAND ==>                                     Enter Q for TMS Query

TAPE1 DD Dsname= BAB.FATSIN.JOBT
        Unit  =( TAPE )
        Label =( ,BLP,EXPDT=980000 )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                     5:  -----  6:  -----  7:  -----  8:  -----
                                     9:  ----- 10:  ----- 11:  ----- 12:  -----
                                    13:  ----- 14:  ----- 15:  ----- 16:  -----
                                    17:  ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :
FILES    - Nbr of Physical files to be read: 3
VALIDATE- Verify Input Volume Serial : NO  (no/yes)

Display more FATS Operands?: NO  (yes/no)

```

Option F (VERIFY READ) on the FATS/FATAR Main Menu brings you to this panel, which uses the FATS READ function to verify the readability of data on the specified tape volumes. The volumes to be processed are specified as described for Option A (Certify New Tapes). The default of reading 3 physical files will read one data file on a labeled tape.

**NOTE:** if you are licensed for FATAR, options G and H provide more information about the content of the tape than option F.

## 93.1 CONTINUED . . .

**FATS VERIFY  
(ANALYZE) OF  
TAPES**

```

----- FATS VERIFY (ANALYZE) TAPES -----
COMMAND ==>                                     Enter Q for TMS Query

TAPE1 DD Dsname= BAB.FATSIN.JOBT                      <
        Unit  =( TAPE )
        Label =( ,BLP,EXPDT=98000                      )

VOL      - Volume serial number(s)  1:  -----  2:  -----  3:  -----  4:  -----
                                     5:  -----  6:  -----  7:  -----  8:  -----
                                     9:  ----- 10:  ----- 11:  ----- 12:  -----
                                    13:  ----- 14:  ----- 15:  ----- 16:  -----
                                    17:  ----- 18:  ----- 19:  ----- 20:  -----

MAXVOLN - Maximum number of volumes :
NUMFILES- Number of Logical files to be read: 0      (zero = all files)
VALIDATE- Verify Volume Serial of input tape: NO      (no,yes)
PRINTOPT(L)- Read and Print tape labels ONLY: NO      (yes/no)
            (A)- Print ALL blocks in ALL files : NO      (yes/no)
            (B)- Number of blocks to print for each file processed: -----
FAST ANALYSIS-Quickly print information about all labeled files: ---- (no/yes)

Display more ANALYZE Operands?: NO  (yes/no)

```

Option G (VERIFY ANALYZE) on the FATS/FATAR Main Menu brings you to this panel, which uses the FATS ANALYZE function to invoke FATAR to analyze the specified tape volumes. The volumes to be processed are specified as described for Option A (Certify New Tapes). The PRINTOPT options allow you to print data from the tapes, or to print only the labels from the first file on the tape (to quickly identify the contents of tapes).

Option G will function only if you are licensed for FATAR.

**NOTE:** Option H (EXAMINE) can be used for the same function. Option H must be used if you want to analyze a multi-volume tape set and get a combined report on the entire set. Option G treats each tape volume as an independent tape and produces a separate report for each. For a single tape volume, use option H.

## 93.2 FATAR ISPF INTERFACE

**FATAR  
OPERATIONS**

Options H through O on the FATS/FATAR main menu will generate jobstreams for FATAR operations and will work only if you are also licensed for FATAR.

Once you enter all necessary options on a given panel (except for option M which is described later), press ENTER to complete generation of the FATAR batch jobstream. You will be given the option of submitting the jobstream immediately or entering ISPF EDIT to view and modify the jobstream; you can submit it directly from edit, or save it in a JCL library for future use.

All the panels contain a dataset name, unit name, and label parameter. A unique default dataset name may be generated but you may overwrite it with another name (this may be required to meet security restrictions). If you are reading labeled tapes (LABEL=SL or AL) you must enter the actual dataset name recorded on the tape label, but if you are bypassing labels (LABEL=BLP) any valid name may be used. For output tapes, the dataset name you specify may be overridden by FATAR. The unit name should be overridden with a generic name (such as 3490) or esoteric name (such as CART) which will allocate a tape drive of the appropriate type; a specific unit name (such as 3A0) may also be used.

All panels contain those FATAR options which are commonly used with the function on that panel; you can override the defaults displayed as necessary. They also contain a question "Display more FATAR Operands?" If you enter YES, the FATAR DEFAULT CONTROL OPERANDS panel shown in Section 93.0 is displayed, where you can override less commonly used parameters. The overrides are effective for this function only; the values are not saved.

**EXAMINE  
CONTENTS  
AND LAYOUT  
(MAP) OF  
TAPES**

```

----- FATAR EXAMINE (MAP) TAPES -----
COMMAND ==>                                     Enter Q for TMS Query

TAPEIN DD Dsname= BAB.FATARIN.JOBT                <
          Unit  =( TAPE )
          Label  =( ,BLP,EXPDT=98000                )
          VolSer=( 99999T                           )

NUMFILES - Number of Logical files to be read : 0      (zero = all files)
BLP      - By-pass tape label processing      : YES     (yes/no)
VALIDATE - Verify Volser and Expiration Date  : NO      (no/ipt/exp) EXP DAYS: --
NEXTVOL  - Multi-volume tape processing method: JCL     (jcl/tms)
PRINTOPT(L)- Read and Print tape labels ONLY  : NO      (yes/no)
          (A)- Print ALL blocks in ALL files   : NO      (yes/no)
          (B)- Number of blocks to print for each file processed : ----(no/yes)
FAST ANALYSIS- Quickly print information about all labeled files: ----(no/yes)

Display more FATAR Operands?: NO (yes/no)

```

Option H (EXAMINE) on the FATS/FATAR Main Menu brings you to this panel, which invokes FATAR to analyze a tape. If you use the default of BLP, the dataset name and volume serial (volser) are not actually verified against the tape, so any values required to get the proper tape mounted and opened can be used; this allows any tape, including tapes with unknown contents and tapes from outside your installation to be mounted. If you change the label type to SL or AL, the actual dataset name and volume serial must be entered. Multiple volume serials may be entered if they are part of a multi-volume tape set. FATAR will report on the actual volume serial, label type, dataset names, and contents of every file on the tape. The PRINTOPT options allow you to print data from the tapes, or to print only the labels from the first file on the tape (to quickly identify the contents of tapes).

**NOTE:** Option G (Figure 10) can be used for the same function. However, when multiple tape serials are entered on Option G, they are treated as individual tapes, not as part of a tape set, and each tape receives a separate FATAR report. Option H requires that multiple serials be part of a multi-volume tape set. Option H should be used for a single tape volume.

## 93.2 CONTINUED . . .

**MAKE AN  
EXACT IMAGE  
OF A TAPE  
VOLUME**

```

----- FATAR IMAGE COPY -----
COMMAND ==>                               Enter Q, QI or QO for TMS Query
TAPEIN DD  Dsname= BAB.FATARIN.JOBT          <
           Unit  =( TAPE          ),Label=( ,BLP,EXPDT=98000      )
           VolSer=( 99999T          )
TAPEOUT DD  Dsname= BAB.FATAROUT.JOBT         <
           Unit  =( TAPE          ),Label=( ,BLP,EXPDT=98000      )
           VolSer=(          )
NUMFILES - Number of Logical files to be read : 0      (zero = all files)
VALIDATE - Verify Volser and Expiration       : NO      (no/ipt/opt/all) EXP DAYS: __
VERIFY   - Verify readability of Output tape : YES      (yes/no)
OUTDSN   - Check for initialized Output Dsname : NO      (yes/no)
OUTVOL   - Use Input Volser for Mount/Validation: NO      (yes/no)
OUTSER   - Retain Output Volume Serial Number : NO      (yes/no)
COMPACTION- Data compression requested for Output: SYSTEM (system/copy/yes/no)

Display more FATAR Operands?: NO (yes/no)

```

Option J (COPY a tape volume) on the FATS/FATAR Main Menu brings you to this panel, which generates a FATAR job to make an image copy (an exact bit-for-bit copy) of a single tape volume. The submitted job **must** be authorized to use BLP (Bypass Label Processing) since this function is required for image copy. Since BLP is used on TAPEIN, any dsname and volser can be specified, whatever is sufficient to mount the proper input tape. However, if INPUT or ALL is specified for VALIDATE, the volume serial of the input tape will be checked against the serial you provided. The dsname of TAPEOUT will not be used, and the TAPEOUT volume serial is whatever you require to get the proper output tape mounted. The output tape will have the actual volume serial of the input tape after the copy is complete, unless you specify YES for OUTSER.

**IMAGE COPY  
OF MULTIPLE  
TAPES**

```

----- FATS MULTIPLE VOLUMES IMAGE COPY -----
COMMAND ==>                               Enter Q for TMS Query
TAPEIN DD  Dsname= BAB.FATSIN.JOBT          <
           Unit  =( TAPE          ),Label=( ,BLP,EXPDT=98000      )
TAPEOUT DD  Dsname= BAB.FATSOUT.JOBT         <
           Unit  =( TAPE          ),Label=( ,BLP,EXPDT=98000      )
VOL         - Volume serial number(s) 1: ----- 2: ----- 3: ----- 4: -----
                                           5: ----- 6: ----- 7: ----- 8: -----
                                           9: ----- 10: ----- 11: ----- 12: -----
                                           13: ----- 14: ----- 15: ----- 16: -----
                                           17: ----- 18: ----- 19: ----- 20: -----
MAXVOLN - Maximum number of volumes: --
VALIDATE - Verify Volser and Expiration       : NO      (no/ipt/opt/all) EXP DAYS: __
VERIFY   - Verify readability of Output tape : YES      (yes/no)
OUTDSN   - Check for initialized Output Dsname : NO      (yes/no)
OUTVOL   - Use Input Volser for Mount/Validation: NO      (yes/no)
OUTSER   - Retain Output Volume Serial Number : NO      (yes/no)
COMPACTION- Data compression requested for Output: SYST (syst/copy/yes/no)

Display more ANALYZE Operands?: NO (yes/no)

```

Option K (COPY multiple volumes) on the FATS/FATAR Main Menu brings you to this panel, which uses the ANALYZE option of FATS to invoke FATAR to perform an image copy of multiple input tapes to multiple output tapes. The input tapes might comprise a multi-volume tape set, or might be unrelated; since image copy completely copies one input tape to one output tape, the relation of the input tapes is immaterial. The considerations for the TAPEIN and TAPEOUT dataset names are the same as for Option J (Copy a tape volume) and the rules for specifying the input volume serials are the same as for Option A (CERTIFY new tapes). However, the job will call for scratch (PRIVAT) mounts for each output tape; your operator must be prepared to mount the proper output tape for each input tape requested.



## 93.2 CONTINUED . . .

### MAKE A LOGICAL COPY OF ONE OR MORE FILES

```

----- FATAR LOGICAL COPY -----
COMMAND ==>                               Enter Q, QI or QO for TMS Query

TAPEIN DD  Dsname= BAB.FATARIN.JOBT          <
            Unit  =( TAPE                      )
            Label =( ,BLP,EXPDT=98000          )
            VolSer=( 99999T                      )
            Dcb   =( ----- )

TAPEOUT DD  Dsname= BAB.FATAROUT.JOBT         <
            Unit  =( TAPE                      )
            Label =( ,BLP,EXPDT=98000          )
            VolSer=( ----- )
            Dcb   =( ----- )

NUMFILES - Number of Logical files to be read : 0      (zero = all files)
BLP/OUTBLP- By-pass label processing on - Input: YES -- on Output: YES (yes/no)
VALIDATE - Verify Volser and Expiration : NO (no/ipt/opt/all/exp) EXP DAYS: --
NEXTVOL - Multi-volume tape processing method: JCL (jcl/tms)
VERIFY - Verify readability of Output tape : YES (yes/no)
LABELS - Label checking to be done by FATAR : YES (yes/os)
COMPACTION- Data compression request for Output: SYSTEM (system/copy/yes/no)
CAT - Catalogue Output tape : NO (yes/no/recat)
Display more FATAR Operands?: NO (yes/no)

```

Option L (COPY one or more files) on the FATS/FATAR Main Menu brings you to this panel, which uses FATAR to logically copy files from an input tape (or multi-volume tape set) to an output tape (or tape set). If you use the default of LABEL=SL (or AL) on TAPEIN, the actual volume serials of the input tape(s) and the actual dataset name of the **first** dataset on the first volume must be provided. If you change the TAPEIN to LABEL=BLP, any dsname and volume serial(s) can be used; FATAR will recognize the label type of the input tape automatically even with BLP. For TAPEOUT, the LABEL parameter determines the format of the output tape (SL for IBM standard labels, AL for ANSI labels, NL or BLP for unlabeled tape); if the output tape is labeled, FATAR will extract the dataset names of the files on the input tape and change the name of the output files to be the same; the dataset name of TAPEOUT is not used. DCB information can be omitted if the input tape has labels; FATAR will get the file characteristics from those labels. If the full dataset name is available (see Section 21), you can specify YES or RECAT for CAT to catalog the dataset to the output tape (use RECAT if the input datasets are already cataloged).

### RECOVER DATA FROM AN OVERWRITTEN TAPE

```

----- FATAR OVER-WRITTEN TAPE RECOVERY -----
COMMAND ==>                               Enter Q, QI or QO for TMS Query

TAPEIN DD  Dsname= BAB.FATARIN.JOBT          <
            Unit  =( TAPE                      )
            Label =( ,BLP,EXPDT=98000          )
            VolSer=( 99999T                      )
            Dcb   =( ----- )

TAPEOUT DD  Dsname= BAB.FATAROUT.JOBT         <
            Unit  =( TAPE                      )
            Label =( ,BLP,EXPDT=98000          )
            VolSer=( ----- )
            Dcb   =( ----- )

NUMFILES - Number of Logical files to be read : 0      (zero = all files)
OUTBLP - By-pass label processing on Output : YES (yes/no)
VALIDATE - Verify output Volser/Expiration Date: (no/opt)
VERIFY - Verify readability of Output tape : YES (yes/no)
EODFIND - Recover all files past 1st EOD mark : -- (no/yes)
EODRETRY - Nbr of attempts to position past EOD: 0 --
COMPACTION- Data compression request for Output: SYST (syst/copy/no/yes)

Display more FATAR Operands?: NO (yes/no)

```

Option O (COPY one or more files) on the FATS/FATAR Main Menu brings you to this panel, which uses FATAR to recover data from a tape which has been overwritten with a smaller amount of data; the data beyond the overwrite can be recovered. The parameters are similar to Option L (COPY). However, BLP is required to position the tape to the beginning of the overwritten data so the submitted job must be authorized to use BLP. A label number **must** be specified, pointing immediately after the overwriting data; since a single labeled dataset uses 4 physical files, LABEL=5 is usually appropriate (see Section 24.2 for more guidance on recovering overwritten data).

## 93.3 FATAR EXTENDED FUNCTIONS ISPF INTERFACE

**FATAR  
EXTENDED  
FACILITIES**

Option M (FATAR extended functions) on the FATS/FATAR Main Menu brings you to this panel, which is a starting menu for an interactive dialog which can build a FATAR jobstream using the many powerful features of FATAR. You should familiarize yourself with the FATAR modification/scan control statements and their functions (Section 23.3) before attempting to use these panels.

Option 1 can be invoked one time to specify the input and optional output tapes and generate the JCL (actually, if you invoke any of the other options before Option 1, it will take you through Option 1 first). The other options, 2-6, can be invoked repeatedly to generate whatever set of FATAR functions are required in that FATAR step.

```

----- FATAR EXTENDED FUNCTIONS -----
OPTION ==>

1. - to ANALYZE data on tape
2. - to DROP (Delete) selected blocks or records
3. - to KEEP selected blocks or change block length
4. - to PRINT selected blocks
5. - to REPLACE (Search for) data in selected blocks or records
6. - to SCAN data fields in selected blocks or records
7. - to RENAME file(s) copied to the output tape

```

As you invoke the various options, this menu will change format to show you how many statements of each kind have been generated. When complete, you can submit the generated job directly, or enter ISPF EDIT to review and submit or save it.

```

----- FATAR EXTENDED FUNCTIONS -----
OPTION ==>

1. - to ANALYZE data on tape                                COUNT
2. - to DROP (Delete) selected blocks or records            1
3. - to KEEP selected blocks or change block length          2
4. - to PRINT selected blocks                                1
5. - to REPLACE (Search for) data in selected blocks or records
6. - to SCAN data fields in selected blocks or records
7. - to RENAME file(s) copied to the output tape

Enter END to Review generated JCL          Enter SUBMIT to Submit Job

```

## 93.3 CONTINUED . . .

**TAPEIN AND  
TAPEOUT  
SPECIFICATIONS**

```

----- FATAR INPUT FILE DEFINITION for ANALYZE OPERATION -----
COMMAND ==>                                     Enter Q for TMS Query

TAPEIN DD Dsname=  BAB.FATARIN.JOBB                               <
          Unit  =( TAPE )
          Label =( , SL )
          VolSer=( 99999B )
          Dcb   =( ----- )

COPYFILE - Do you wish to create an Output Tape?:  __ (yes/no)

BLP      - By-pass labels processing on Input : YES      (yes/no)
VALIDATE - Verify Volser/Expdt of Input tape : NO       (no/ipt/exp)  EXP DAYS:
LABELS   - Label checking to be done by FATAR : YES     (yes/no/os/image/req)
NUMFILES - Number of Logical files to be read : 0       (zero = all files)
NEXTVOL  - Multi-volume tape processing method: JCL     (jcl/tms)
EODRETRY - Attempts to position past EOD : 0
ENDAFTR(LF) - Stop FATAR after logical file : -----
          (B) - Stop FATAR after block number : -----

Display more FATAR Operands?: NO (yes/no)

```

Option 1 will take you to this screen to specify the input tape (you also come here if you enter another option without using Option 1 first). Refer to Option L (COPY one or more files) for considerations for the input (and output) tape specification.

If you specify YES for COPYFILE, the next panel allows you to specify the output tape, including the ability to optionally catalog the output files.

```

----- FATAR OUTPUT FILE DEFINITION for ANALYZE OPERATION -----
COMMAND ==>                                     Enter Q for TMS Query

TAPEOUT DD Dsname=  BAB.FATAROUT.JOBB                               <
          Unit  =( TAPE )
          Label =( ,BLP,EXPDT=98000 )
          VolSer=( ----- )
          Dcb   =( ----- )

OUTBLP   - By-pass label processing on Output : YES      (yes/no)
VALIDATE - Verify Volser/Expiration of Output : NO       (no/opt)
COMPACTION- Data compression request for Output : SYST   (syst/copy/yes/no)
COMBFILES - Combine all input into 1 Output file : NO     (yes/no)
VERIFY   - Verify readability of Output tape : YES      (yes/no)
CAT      - Catalogue Output tape : NO                   (yes/no/recat)
OUTDSN   - Check for initialized Output Dsname : NO      (yes/no)
OUTVOL   - Use Input Volser for Mount/Validation: NO     (yes/no)
OUTSER   - Retain Output Volume Serial Number : NO      (yes/no)

Display more FATAR Operands?: NO (yes/no)

```

## 93.3 CONTINUED . . .

**FATAR DROP  
OPERATION**

```

----- FATAR DROP OPERATION #2 -----
COMMAND ==> _
Drop Data Check Block ONLY: N (Y/N)
Logical File..... ALL: N (Y/N)    -OR-    File: _____ Hdr/Trl: __ (H/T)
Blocks/Records..... ALL: N (Y/N)    -OR-    Blk1: _____ Rcd1: _____
                                           Blk2: _____ Rcd2: _____

```

This is Option 2 on the FATAR Extended Functions Panel, which allows you to drop whole blocks or logical records from the output tape. See Section 23.3.2 for guidance on the DROP statement.

**FATAR KEEP  
OPERATION**

```

----- FATAR KEEP OPERATION #1 -----
COMMAND ==> _
Keep Data Check Block ONLY: N (Y/N)
Logical File..... ALL: N (Y/N)    -OR-    File: _____ Hdr/Trl: __ (H/T)
Blocks..... ALL: N (Y/N)    -OR-    Blk1: _____ Blk2: _____
New Block Length..... : _____
Starting Location..... : _____
Reposition Data..... : L (L -- Justify Left)
                      (R -- Justify Right)

```

This is Option 3 on the FATAR Extended Functions Panel, which allows you to copy whole blocks or logical records to the output tape when they would otherwise be dropped. It can also modify the length of copied blocks. See Section 23.3.3 for guidance on the KEEP statement.

**FATAR PRINT  
OPERATION**

```

----- FATAR PRINT OPERATION #1 -----
COMMAND ==> _
Print ALL Blocks: N (Y/N)
----- OR -----
Logical File..... ALL: N (Y/N)    -OR-    File: _____ Hdr/Trl: __ (H/T)
Blocks..... ALL: N (Y/N)    -OR-    Blk1: _____ Blk2: _____
Data Print Length..... : _____
Data Print Format..... : _____ (Char/Hex/Dump)

```

This is Option 4 on the FATAR Extended Function Panel, which allows you to print blocks or records from the input tape. See Section 23.3.1 for guidance on the PRINT statement.

## FATAR REPLACE OPERATION

```

----- FATAR REPLACE OPERATION #1 (Panel 1 of 2) -----
COMMAND ==> _

Replace Data Check Block ONLY: N (Y/N)
Logical File..... ALL: _ (Y/N)      -OR-      File: _____ Hdr/Trl: __ (H/T)
Blocks/Record..... ALL: _ (Y/N)      -OR-      Blk1: _____ Rcd1: _____
                                           Blk2: _____ Rcd2: _____

Starting Location..... : _____

Scan until End of Rcd/Blk... : N (Y/N)

Scan Option..... : _ (S - for Search)
                  (V - for Verify)

Replace Data Option..... : N (Y/N)

```

This is Option 5 on the FATAR Extended Function Panel, which allows you to scan for data in blocks or records on the input tape and optionally replace data in blocks or records written to the input tape. See Section 23.3.4 for guidance on the REPLACE statement.

```
----- FATAR REPLACE OPERATION #1 (Panel 2 of 2) -----  
COMMAND ==> _  
  
SEARCH STRING -      (* Note: Character Type: C for Ebcdic, X for Hexadecimal )  
Type:              <  
-                  <  
-                  <  
-                  <  
-                  <  
-                  <  
REPLACEMENT DATA --  
Type:              <  
-                  <  
-                  <  
-                  <  
-                  <  
-                  <  
->>>>> Enclose your character string(s) in Quotes (ex: 'xx...xx')    <<<<<
```

This is the second panel for the REPLACE function, where the data to be located and the replacement data are entered as either character or hexadecimal strings. The panel may contain either the SEARCH STRING field, the REPLACEMENT DATA field, or both, depending on options entered on the first panel.

## FATAR SCAN OPERATION

```

----- FATAR SCAN OPERATION #1 -----
COMMAND ==> _

Scan Data Check Block ONLY: N (Y/N)
Logical File.....ALL : _ (Y/N) -OR- File: ____ Hdr/Trl: __ (H/T)

Blocks/Record.....ALL : _ (Y/N) -OR- Blk1: _____ Rcd1: _____
                                           Blk2: _____ Rcd2: _____

Length of Field.....: _____
Starting Location.....: _____
Scan Type.....: --(P-- scan for Packed decimals)
                  (Z-- scan for Zoned decimals)
                  (t-- scan according to Table <t>:
                        where <t> = 'A' to 'Y' <excl. 'P'>))

```

This is Option 6 on the FATAR Extended Function Panel, which allows you to validate data on the input tape; you can check for valid packed or zoned decimal fields, or validate against a table of valid bytes you provide. See Section 23.3.5 for guidance on the SCAN statement.

## FATAR TABLE DEFINITION

If you specify a Table Name other than P or Z on the previous panel, this panel allows you to specify the characters that are valid (or invalid) for the field to be validated. See Section 23.3.6 for guidance on the TABLE statement.

This is Option 7 on the FATAR Extended Function Panel, which allows you to rename output datasets (if you specified an output tape); you can check for valid packed or zoned decimal fields, or validate against a table of valid bytes you provide. See Section 23.3.7 for guidance on the RENAME statement.

**READER'S COMMENT FORM**  
**FATS/FATAR USERS MANUAL V4.8**  
**INNOVATION DATA PROCESSING**

If you have comments on this manual, including:

- errors in the text or typographical errors
- clarity
- suggestions for improvement in the manual
- suggestions for improvement in the product
- any other comments

Please complete this form and fax it to Innovation at 973-890-7147 (in Europe you may fax it to your local Innovation office as shown on the front page of the manual). You may also e-mail your comments to Innovation at [support@fdrinnovation.com](mailto:support@fdrinnovation.com) (be sure to identify the manual name in the message).

Your name: \_\_\_\_\_

Company name: \_\_\_\_\_

Mailing address: \_\_\_\_\_

\_\_\_\_\_

E-mail address: \_\_\_\_\_

Comments: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**